METHODS OF ITERATIVE IMPROVEMENT FOR THE NUMERICAL SOLUTION OF OPERATOR EQUATIONS

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

By

D. DANIEL SATHIARAJ

to the

COMPUTER SCIENCE PROGRAMME
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
DECEMBER, 1982

TO

my Heavenly Father,
the Wonderful Counsellor

and

my Saviour,
the King of kings

CSP-1982-D-SAT-MET

CERTIFICATE

This is to certify that the research work embodied in the thesis 'METHODS OF ITERATIVE IMPROVEMENT FOR THE NUMERICAL SOLUTION OF OPERATOR EQUATIONS' by D. Daniel Sathiaraj has been carried out under my supervision, and that this work has not been submitted elsewhere for a degree.

R. Sankar

R. Sankar
Professor of Computer Science
Indian Institute of Technology
Kanpur

ACKNOWLEDGEMENTS

I wish to express my gratitude to my supervisor Professor R. Sankar for his personal interest, able guidance and constant encouragement throughout my doctoral programme.

I am thankful to the management of St John's College, Palayamkottai for granting me study-leave for my Ph.D. degree programme, and to Professor P.C.Sood, the Director of Computer Centre, Banaras Hindu University, Varanasi for permitting me to use the computing and research facilities during my tenure as Lecturer in Computer Science since July 1981.

I would like to thank the many people who were interested in the completion of my thesis, and those who helped me during the final stage of this work. I acknowledge with thanks the careful typing of the text, which was done by M. Anwar, and the cyclostyling of the stencils by R.Y.Misra.

My special thanks go to my late father-in-law for his encouragement and to my wife Pansy for her patient love amidst the strain placed upon my family by my research.

D. Daniel Sathiaraj

CONTENTS

SYNOPSIS

METHODS OF ITERATIVE IMPROVEMENT FOR THE NUMERICAL SOLUTION OF OPERATOR EQUATIONS

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY
by

D. DANIEL SATHIARAJ
Computer Science Programme
Indian Institute of Technology,Kanpur

November 1982

Until recently there have been three standard iterative techniques for improving the accuracy of numerical solutions: Richardson extrapolation, Iterated Deferred Correction (IDC) of Fox and Pereyra, and the relatively new Iterated Defect Correction (IDeC). The method IDeC was originated by Zadunaisky as error estimation, and later formulated by Stetter for iterative application. The present thesis introduces new and generalized methods of iterative improvement and presents a variant of IDeC as well as various novel applications of IDeC.

A historical survey outlining the related developments is given in the introductory chapter of this thesis. Chapter 2 deals with new applications of IDeC based on quadrature methods for the solution of linear Fredholm and linear and nonlinear Volterra integral equations of the second kind as well as of linear eigenvalue problems.

Chapters 3 to 6 present four new acceleration techniques which have been named as Successive Extrapolated IDeC (SEIDeC), Iterated Deferred Defect Correction (IDDeC), Successive Extrapolated IDDeC (SEIDDeC), and IDeC on Extrapolation (IDeCE). These methods, which involve IDeC with the use of interpolating polynomials, are described as applied to quadrature methods of Fredholm integral equations. The asymptotic expansions for the methods are derived; implementation details are provided; and illustrative numerical examples have been worked out. Applications to linear and nonlinear Volterra equations have also been considered.

A variant of IDeC called Successive Updated IDeC (SUIDeC) is presented in Chapter 7. A comparative study of all these acceleration techniques is made in Chapter 8. The ninth chapter is devoted to applications of defect correction,with the use of splines, to different types of partial differential equations.

The class of acceleration techniques has been enriched remarkably by the IDeC method. When applied to Volterra equations, it produces solutions of high accuracy even for relatively large stepsizes. For Fredholm equations with smooth kernels and functions, the easy-to-implement SEIDeC method extends the order of convergence arbitrarily beyond the maximum order attainable by IDeC; each iteration of IDeC and IDDeC increases the order of convergence by n and n+p respectively, where n is the order of the basic discretization scheme and p is the increase in order due to deferred correction. Because of its guaranteed convergence, IDDeC is superior to IDC also. The method of SEIDDeC extends the order of convergence of IDDeC.

Numerical results confirm theoretical analysis and indicate the capability of the defect correction techniques in providing general software for the automatic computation of solutions to problems not only in integral equations but also in other operator equations.

All the above methods have general applicability and will be useful in various areas. The computations have been carried out using DEC 1090 at IIT,Kanpur and ICL 1904S at BHU, Varanasi.

CHAPTER 1

INTRODUCTION

In this thesis we discuss a number of techniques for improving the accuracy of the numerical solution of operator equations of the form

$$F(y) = 0 \tag{1.1}$$

where the function y is the exact solution.

In most of the discretization methods which are used for solving equation (1.1) the numerical solution at the discrete points converges to the true solution as the discretization parameter h tends to zero. However the rate of convergence is often slow, and for satisfactory accuracy it becomes necessary to choose a very small h. Hence we need techniques to accelerate this rate of convergence or equivalently for procedures which estimate the error in the numerical solution with a view to improve the accuracy in the solution iteratively. During the past century a number of such techniques have been introduced in the literature.

Since the beginning of this century, asymptotic expansions of the global error in powers of h have been used to construct methods of increasing accuracy. One of the methods is that of Richardson[1910, 1927] called by him the 'deferred approach to the limit'. In this method, solutions of increasing accuracy are produced on the original grid by the use of solutions on finer grids.

Another well-known and widely applicable acceleration technique is the 'difference correction' or 'deferred correction' method of Fox[1947, 1949, 1950, 1957]. This method performs computations on the original grid of discretization itself without refining the grid but uses high order approximations without increasing the complexity of the linear or nonlinear system resulting from the discretization scheme for (1.1). This method has later been analyzed, refined and extended by Pereyra[1966, 1967a, 1967b, 1968, 1970], who named the refined version of Fox's method as 'Iterated Deferred Correction' (IDC). Pereyra used the results on the asymptotic expansions for the global error, such as in Gragg[1965] and Stetter[1965], and showed that for sufficiently small h each iteration of IDC improves the order of accuracy by p, where p is the order of the discretization scheme. The IDC method gives much better and faster results than most other methods of discretization in real practical problems (see Fox[1977]).

Recently an interesting and efficient approach to global error estimation has been developed by the Argentinian astronomer P.E. Zadunaisky. This new approach discussed by Lambert[1977] and the related method of Iterated Defect Correction by Stetter[1974, 1978] and Frank[1976, 1977] stimulated the work presented in this thesis. Our primary goal was to apply Zadunaisky's device iteratively in new ways and in new types of operator equations other than ordinary differential equations.

In the next section of this chapter historical outlines of the development of Zadunaisky's principle and the method of Iterated Defect Correction are given. The relationships between this method and other methods of iterative improvement, such as those introduced by Frank, Hertling and Ueberhuber and by Lindberg, are also pointed out. A summary of the thesis is given in Section 1.2.

1.1 Historical Survey

Zadunaisky, when solving certain problems in Astronomy which involved highly accurate observations found it impossible or very difficult to apply the then known methods for the purpose of achieving the desired accuracy in the numerical solution of a system of ordinary differential equations (Zadunaisky[1976]). So he was motivated by the desire to devise a good procedure for the estimation of the global errors propagated in the solution, and thereby to improve the accuracy of that solution.

He successfully developed one such procedure and introduced it first in a symposium of the International Astronomical Union in 1964 (Zadunaisky[1966a]). Though he had also discussed this procedure in his subsequent papers (Zadunaisky[1966b, 1970, 1972]) which deal with problems in astronomy, the first time a group of numerical analysts came to know of his idea was at the Dundee Conference on Numerical Analysis in 1973 when Zadunaisky presented a paper. In that paper he had included a mathematical description of his procedure, and impressive numerical results for problems in systems of differential equations, two-point boundary value problems, and application to linear multistep and predictor-corrector methods.

Stetter[1974] subsequently found theoretical results justifying Zadunaisky's numerical results. He formalized Zadunaisky's idea of error estimation and conceived of its iterative use.

The power of Zadunaisky's approach is indicated in the mathematical analysis of Frank[1975] on the asymptotic behaviour of the approach when applied to Runge-Kutta methods of order p. He proved that under suitable smoothness conditions Zadunaisky's estimate for global error differs from the actual global error by $O(h^{2p})$.

Alfeld[1975] gives a survey of Zadunaisky's device applied to initial and boundary value problems of ordinary differential equations. The papers of Frank[1976, 1977] deal with iterative application of Zadunaisky's concept of error estimation to two-point boundary value problems; he called this technique the method of 'Iterated Defect Correction' (IDeC), whereas Stetter[1974] had introduced the term Differential Correction in place of Defect Correction. Moreover, Frank was the first to prove rigorous results concerning the asymptotic behaviour of IDeC applied to classical discretization methods for two-point boundary value problems.

Frank and Ueberhuber[1975] proved that for an IDeC procedure based on an explicit or implicit Runge-Kutta method of order p the sth iterate is of order s.p, the maximum attainable order of the procedure being limited by the degree of polynomials used in piecewise interpolation of the discrete solution. Besides, their IDeC applications, on the implicit Euler scheme, have proved to be efficient in the solution of stiff systems of ordinary differential equations (Frank and Ueberhuber[1977]).

Meanwhile, by adapting Zadunaisky's idea Frank, Hertling and Ueberhuber[1976, 1977] introduced a new technique for estimating local discretization errors and a new variant to IDC methods based on that technique. Unlike the other IDC methods, these IDC methods do not depend on explicit knowledge of the asymptotic expansion for the local discretization error. Consequently, they have a larger domain of applicability when compared with other IDC implementations such as those of Pereyra[1973] and Lentini and Pereyra[1975], in which the computation of deferred corrections is rather difficult and is reliable only for suitable choices of the discretization parameter h (see Daniel and Martin[1977]).

Stimulated by Zadunaisky's approach discussed in Stetter[1974], Lindberg[1976] developed yet another technique for iterative improvement. This technique involves a sequence of perturbed problems but is quite different from that of Zadunaisky. It is also more general than the difference correction procedure of Fox[1957]. In Lindberg's technique local error estimates are constructed avoiding detailed local error analysis, whereas in Pereyra's method it is necessary to use local error expansions. Lindberg points out a disadvantage of Pereyra's version of IDC, namely that 'when F(y) is nonlinear in some derivative of y, the expansions are very cumbersome to find and even more cumbersome to approximate'. Lindberg also gives various examples for the application of his technique to ordinary and partial differential equations and to integral equations. He shows that for each iteration the order of accuracy of the new approximate solution is increased as in the case of IDC.

The notable difference between the IDeC method and Lindberg's technique is that in the former a smooth global approximation to the solution of the operator equation $F(y) = 0$ is constructed as an extension of the discrete solution and then perturbation or defect is computed from that approximation, whereas the latter technique uses the discrete solution directly and the local properties of the perturbations.

In his survey paper, Stetter[1978] exposes the general structure of different techniques for iterative improvement based on deferred correction or defect correction, and also proves the convergence of iterates of the IDeC method towards a fixed point using contraction principle but avoiding asymptotic expansions.

As a continuation of Lindberg's work Skeel[1980] discusses theoretical aspects of iterated deferred correction avoiding the asymptotic expansions of the global error.

In the light of the past research outlined above, the present thesis introduces new and generalized methods of iterative improvement and presents a variant of IDeC as well as various novel applications of IDeC.

1.2 Summary of the Thesis

The next chapter deals with new applications of IDeC based on quadrature methods for the solution of linear Fredholm, and linear and nonlinear Volterra integral equations of the second kind. We also bring out the difficulties encountered in the application of IDeC to eigenvalue problems.

Chapters 3 to 6 present four new acceleration techniques which have been named as Successive Extrapolated IDeC (SEIDeC), Iterated Deferred Defect Correction (IDDeC), Successive Extrapolated IDDeC (SEIDDeC), and IDeC on Extrapolation (IDeCE). These methods, which involve IDeC with the use of interpolating polynomials, are described as applied to quadrature methods of Fredholm integral equations. The asymptotic expansions for the methods are derived; implementation details are provided; and illustrative numerical examples have been worked out. Applications to linear and nonlinear Volterra equations have also been considered.

A variant of IDeC called Successive Updated IDeC (SUIDeC) as applied to nonlinear Volterra equations is presented in Chapter 7. A comparative study of all these acceleration techniques is made in Chapter 8. The ninth chapter is devoted to applications of defect correction, with the use of splines and polynomials, to different types of partial differential equations. In the last chapter conclusions are drawn and future research possibilities are discussed.

The class of acceleration techniques has been enriched remarkably by the IDeC method. When applied to

Volterra equations, it generally produces solutions of high accuracy even for relatively large stepsizes. For Fredholm equations with smooth kernels and functions, the easy-to-implement SEIDeC method extends the order of convergence arbitrarily beyond the maximum order attainable by IDeC; each iteration of IDeC and IDDeC increases the order of convergence by n and n+p respectively, where n is the order of the basic discretization scheme and p is the increase in order due to deferred correction. Because of its guaranteed convergence, IDDeC is superior to IDC also. The method of SEIDDeC extends the order of convergence of IDDeC

Numerical results confirm theoretical analysis and indicate the capability of the defect correction techniques in providing general software for the automatic computation of solutions to problems not only in integral equations but also in other operator equations.

The thesis has been written in the accepted mathematical style of establishing certain theoretical results and giving numerical confirmation by specific examples. It is well-known that most theoretical results are guided by plausible reasoning and numerical results indicating the plausibility. Chronologically, the work presented in this thesis had gone through the plausible arguments, numerical computations and theoretical justifications. The details are given below.

(i) Having found the success of Zadunaisky's method in ordinary differential equations, it looked highly plausible that it should work in partial differential equations as well. As such our initial work is what is contained in Chapter 9. However, due to serious limitations in our being able to create satisfactory bivariate function to compute the defect we could not complete this problem. So we attempted solutions by a series of univariate interpolatory functions but without much numerical success.

(ii) We then turned our attention to applications of IDeC to integral equations. We found that the use of interpolatory polynomials is preferable to that of interpolatory cubic splines, because the former has a smaller interpolation error.

(iii) The derivation of theoretical results as well as the construction of new acceleration techniques described in this thesis were motivated by an analysis of numerical results obtained by IDeC methods for integral equations.

CHAPTER 2

APPLICATIONS OF IDeC TO INTEGRAL EQUATIONS

2.1 Introduction

In the area of application of the method of iterated defect correction (IDeC) to ordinary differential equations, substantial research has been done by Alfeld (1975), Zadunaisky (1976), the group led by Frank (1975-1977), Stetter (1979), and Zadunaisky and Lafferriere (1980). However, there have been only a few or no applications of IDeC to other types of operator equations such as partial differential equations and integral equations.

In this chapter we deal with applications of the relatively new IDeC method to integral equations. We consider IDeC techniques based on quadrature methods for the numerical solution of Fredholm and Volterra integral equations of the second kind.

The next section contains a description of the method of IDeC as applied to linear Fredholm integral equations of the second kind, and illustrates the efficiency of IDeC by means of numerical examples. Similarly Sections 2.3 and 2.4 deal respectively with non-linear and linear Volterra integral equations of the second kind. The difficulties involved in the application of IDeC to eigenvalue problems are brought out in Section 2.5.

2.2 Fredholm Integral Equations of the Second Kind

Consider a linear Fredholm integral equation (FIE) of the second kind:

$$(2.2.1)\quad y(x) - \int_b^a k(x,t)\, y(t)\, dt = f(x), \qquad a \leq x \leq b.$$

In this equation the kernel $k(x,t)$ is a known function, $f(x)$ is also known, and the function $y(x)$ is to be determined. We assume that $k(x,t)$ is well-behaved for any x and t in $[a,b]$ and $f(x)$ is sufficiently smooth so that a unique solution $y(x)$ exists. An approximate solution to $y(x)$ can be found as described below.

Let N be a positive even integer, $h = (b-a)/N$, and $x_i = t_i = a + ih$, $i = 0(1)N$. We then use a Newton-Cotes quadrature formula such as Simpson's rule to replace the integral in (2.2.1) by a sum. The substitution $x = x_i$ in the resulting equation leads to the equation

$$(2.2.2)\quad Y_i - \sum_{j=0}^{N} w_j\, k(x_i,t_j)\, Y_j = f(x_i), \qquad i = 0(1)N,$$

where Y_i is an approximation to $y(x_i)$, and the w_j are the weights in the quadrature formula. The system of simultaneous linear algebraic equations (2.2.2) may then be solved for the determination of the Y_i by Gaussian elimination. The above method is called a quadrature method and is denoted by Quad(h).

We improve these approximations Y_i by applying the method of Iterated Defect Correction (IDeC) as explained in the next section.

2.2.1 The Method of Iterated Defect Correction

We first estimate the unknown global error $y(x_i) - Y_i$ in the numerical solution Y_i. Assume that the integer N can be expressed as a product of two integers m and s, where m is a fixed even integer greater than two.

We divide the interval $[a,b]$ into subintervals $I_1, I_2, \ldots, I_s$, where $I_j = [x_{(j-1)m}, x_{jm}]$, $j = 1(1)s$. For each j, let $P^{(0)}_{j,m}(x)$ be the polynomial of degree m which interpolates the m + 1 points (x_i, Y_i), $(j - 1)m \leq i \leq jm$.

We then define for x in $[a,b]$, the interpolating function $P^{(0)}_m(x)$ as the collection of polynomials $P^{(0)}_{j,m}$ such that

$$P^{(0)}_m(x) = P^{(0)}_{j,m}(x) \qquad \text{for } x \in I_j, \qquad j = 1(1)s.$$

(We observe that these polynomials depend also on the stepsize h even though the notation does not bear out this fact.)

We now construct a new problem

$$(2.2.3a) \quad y(x) - \int_a^b k(x,t)\, y(t)\, dt = g^{(0)}(x), \qquad a \leq x < b$$

where

$$(2.2.3b) \quad g^{(0)}(x) = P_m^{(0)}(x) - \int_a^b k(x,t)\, P_m^{(0)}(t)\, dt.$$

Since $g^{(0)}(x) - f(x)$ is small, (2.2.3a) is a neighbouring problem (NP) of the original problem (OP) given by (2.2.1). The exact solution of this NP is the known function $P_m^{(0)}(x)$. We then solve this problem (2.2.3a) by the same method Quad(h), and denote its solution at x_i by $\eta_i(0)$, $i = 0(1)N$. The error in this solution is known, and is equal to $P_m^{(0)}(x_i) - \eta_i^{(0)}$.

Since both OP (2.2.1) and NP (2.2.3a) are solved by the same method, we postulate that the <u>unknown</u> error $y(x_i) - Y_i$ is approximately equal to the <u>known</u> error $P_m^{(0)}(x_i) - \eta_i^{(0)}$. Therefore, the approximation $Y_i^{(1)}$ given by

$$(2.2.4a) \qquad Y_i^{(1)} = Y_i + P_m^{(0)}(x_i) - \eta_i^{(0)}$$

is a better approximation to $y(x_i)$ than Y_i. This principle of error estimation is due to Zadunaisky (1976).

The process above may be repeated as many times as are necessary. The improved approximations $Y_i^{(r+1)}$ are then given by

$$(2.2.4b) \quad Y_i^{(r+1)} = Y_i + (P_m^{(r)}(x_i) - \eta_i^{(r)}), \qquad r = 1,2,\ldots,$$

where the function $P_m^{(r)}(x)$, defined analogous to the $P_m^{(0)}(x)$ of (2.2.3a), interpolates the points $(x_i, Y_i^{(r)})$, $i = 0(1)N$ and the $\eta_i^{(r)}$ is the numerical solution of the neighbouring problem

$$(2.2.5a) \quad y(x) - \int_a^b k(x,t)\, y(t)\, dt = g^{(r)}(x), \qquad a \leq x \leq b$$

where

$$(2.2.5b) \quad g^{(r)}(x) = P_m^{(r)}(x) - \int_a^b k(x,t)\, P_m^{(r)}(t)\, dt$$

for which the exact solution is $P_m^{(r)}(x)$. This technique is called the method of Iterated Defect Correction (Sathiaraj and Sankar[1982]).

We observe that, for each $r = 0,1,2,\ldots$, $\eta_i^{(r)}$ is obtained as the solution of a new system of N+1 equations

$$(2.2.6) \qquad \eta_i^{(r)} - \sum_{j=0}^{N} w_j k(x_i,t_j)\eta_j^{(r)} = g^{(r)}(x_i)$$

This system has the same matrix of coefficients as the system (2.2.2), but the term $g^{(r)}(x_i)$ replaces the right hand side $f(x_i)$ of (2.2.2). Consequently, when finding the solution Y_i of (2.2.2), if we represent the matrix of co-efficients in the product form LU, where L and U are triangular matrices, the Gaussian elimination need be done only once. This requires $O(N^3)$ operations, whereas the determination of the solution $\eta_i^{(r)}$ of (2.2.6) requires only $O(N^2)$ operations, if the computation of the right-hand side of (2.2.6) is ignored.

Implementation

When solving (2.2.6) for the numerical solution $\eta_i^{(r)}$ of the neighbouring problem (2.2.5a) with a fixed r, we need to compute the values $g^{(r)}(x_i)$, $i = 0(1)N$. By the definition of $P_m^{(r)}(x)$, we have from (2.2.5b)

$$(2.2.7) \qquad g^{(r)}(x_i) = Y_i^{(r)} - \int_a^b k(x_i,t)\; P_m^{(r)}(t)\; dt.$$

The integrand of the integral in (2.2.7) is a known function, and therefore the integral may conveniently be computed as accurately as necessary. In our computations, NAG routines were used for this purpose.

2.2.2 Numerical results

In this section the relative performance of the methods of quadrature and iterated defect correction is illustrated.

The problems below were solved using 16-18 significant digits. The first four of these problems are taken from Netravali and de Figueiredo (1974) and the last problem from Baker (1977).

Problem 1

$$y(x) - \int_0^1 xt\, y(t)\, dt = e^x - x$$

exact solution: $y(x) = e^x$.

Problem 2

$$y(x) - \int_0^1 xt\, y(t)\, dt = \sin \pi x - x/\pi$$

exact solution: $y(x) = \sin \pi x$

Problem 3

$$y(x) - \int_0^1 x^4\, e^{xt}\, y(t)\, dt = x - x^2 + x^2\, e^x(1 - x)$$

exact solution: $y(x) = x$

Problem 4

$$y(x) - \int_0^1 x^4\, e^{xt}\, y(t)\, dt = \sin \pi x - \frac{\pi x^4(e^x + 1)}{(x^2 + \pi^2)}$$

exact solution: $y(x) = \sin \pi x$

Problem 5

$$y(x) - \int_0^1 e^{xt}\, y(t)\, dt = 1 - (e^x - 1)/x$$

exact solution: $y(x) = 1$

The methods were applied using both the trapezoid and Simpson's rules. The degrees of the interpolating polynomials was chosen to be 6, 8, 10 or 12, and the number N of intervals in [0,1] to be m, 2m and 4m. In Tables 2.2.1 to 2.2.5 are listed the maximum values of the absolute error in the computed solution for the relevant cases.

We denote the IDeC methods based on the trapezoid rule and Simpson's rule respectively by IDeC_T and IDeC_S.

In our tables, the results for the quadrature methods based on the trapezoid and Simpson's rule appear as entries corresponding to the iteration number 0 of the corresponding IDeC methods. An entry with an asterisk (*) indicates that the subsequent iterates also have the same entry.

The results for Problems 1 and 3 by IDeC_T with m = 10 and 12 are given in Table 2.2.1, and those for problems 1 to 4 by the IDeC_S in Table 2.2.2. We find that both the IDeC methods work much better than the underlying quadrature methods and that IDeC_S has a more rapid convergence than IDeC_T. With a single defect correction, IDeC_S is capable of producing a highly accurate solution.

In Table 2.2.3, we display the results for the IDeC_T and IDeC_S methods with m = 8, and in Table 2.2.4 those for the methods with m = 6. We notice that the iterates of both the IDeC methods converge to the same solution, and that IDeC_S needs generally just one iteration to reach the point of convergence.

Table 2.2.1 Maximum errors-IDeC Method on Trapezoid Rule

			Iteration Number						
	m	N	0	1	2	3	4	5	6
Problem 1	10	10	5.6E-3	1.4E-5	3.5E-8	8.7E-11	2.3E-13	1.2E-14	1.3E-14
		20	1.4E-3	8.7E-7	5.4E-10	3.5E-13	2.1E-16	1.7E-18	1.3E-18
		40	3.5E-4	5.4E-8	8.5E-12	1.3E-15	3.0E-18	2.0E-18	8.7E-18
	12	12	3.9E-3	6.7E-6	1.2E-8	2.0E-11	3.5E-13	8.3E-17	9.5E-18
		24	9.6E-4	4.2E-7	1.8E.10	7.9E-14	4.2E-17	7.8E-18	5.2E-18
		36	2.4E-4	2.6E-8	2.8E-12	3.1E-15	1.1E-17	1.1E-17	7.8E-18
Problem 3	10	10	6.1E-3	1.5E-4	3.5E-6	8.2E-8	1.9E-9	4.5E-11	1.0E-12
		20	1.5E-3	9.2E-6	5.3E-8	3.0E-10	1.7E-12	1.0E-14	5.5E-17
		40	3.7E-4	5.7E-7	8.1E-10	1.2E-12	1.7E-15	4.3E-18	2.4E-18
	12	12	4.2E-3	7.2E-5	1.2E-6	1.9E-8	3.0E-10		
		24	1.0E-3	4.4E-6	1.8E-8	7.0E-11	2.8E-13		
		36	2.6E-4	2.7E-7	2.7E-10	2.7E-13	2.6E-16		

Table 2.2.2 Maximum Errors- IDeC Method Based on Simpson's Rule

			Iteration Number			
	m	N	0	1	2	3
Problem 1	10	10	6.6E-6	1.3E-14*		
		20	4.1E-7	2.6E-18	8.7E-19	
	12	12	3.2E-6	7.8E-18	6.9E-18	
		24	2.0E-7	3.5E-18	4.3E-18	
Problem 2	10	10	2.6E-5	2.7E-10*		
		20	1.6E-6	4.9E-14*		
		40	1.6E-7	1.1E-17	1.0E-17*	
	12	12	1.3E-5	1.5E-12*		
		24	7.8E-7	6.7E-17	6.6E-17	6.7E-17
Problem 3	10	10	6.5E-6	5.9E-9	2.5E-12	1.1E-15
		20	4.0E-7	2.4E-11	6.3E-16	2.2E-18
		40	2.5E-8	9.4E-14	2.6E-18	
	12	12	3.1E-6	1.4E-9	2.8E-13	6.2E-17
		24	2.0E-7	5.5E-12	7.0E-17	8.7E-19
		48	1.2E-8	2.2E-14	6.5E-18	
Problem 4	10	10	8.0E-4	1.5E-8	1.2E-9	1.1E-9*
		20	5.0E-6	6.3E-11	3.6E-13*	
		40	3.1E-7	2.5E-13	8.6E-17*	
	12	12	3.9E-6	3.7E-9	5.3E-12	6.2E-12*
		24	2.4E-6	1.5E-11	3.0E-16	5.2E-16*
		48	1.5E-7	5.7E-14	2.4E-18	2.9E-18

Table 2.2.3 Maximum Errors- IDeC MethodsBased on Trapezoid(T) and Simpson's(S) Ruleswith m=8

			Iteration Number					
Problem	N	Rule	0	1	2	3	4	5
1	8	T	8.7E-3	3.4E-5	1.3E-7	5.1E-10	1.7E-11	1.5E-11
		S	1.6E-5	1.5E-11*				
	16	T	2.2E-3	2.1E-6	2.1E-9	2.0E-12	1.7E-14	1.6E-14
		S	1.0E-6	1.6E-14*				
	32	T	5.4E-4	1.3E-7	3.2E-11	7.9E-15	1.7E-17	1.6E-17
		S	6.3E-8	1.7E-17	1.6E-17	1.5E-17	1.5E-17	
2	8	T	6.2E-3	2.4E-5	1.3E-7	3.9E-8*		
		S	6.4E-5	3.9E-8*				
	16	T	1.5E-3	1.5E-6	1.5E-9	2.7E-11	2.8E-11*	
		S	4.0E-6	2.8E-11*				
	32	T	3.8E-4	9.4E-8	2.3E 11	2.0E-14	2.6E-14*	
		S	2.5E-7	2.6E-14*				
3	8	T	9.6E-3	3.8E-4	1.4E-5	5.1E-7	1.9E-8	5.9E-10
		S	1.6E-5	3.4E-8	3.4E-11	3.7E-14	3.8E-17	2.2E-18
	16	T	2.3E-3	2.3E-5	2.0E-7	1.8E-9	1.6E-11	1.5E-13
		S	9.9E-7	1.4E-10	9.1E-15	2.2E-18	2.0E-18	2.0E-18
	32	T	5.8E-4	1.4E-6	3.1E-9	7.0E-12	1.6E-14	3.6E-17
		S	6.2E-8	5.6E-13	1.3E-18	1.7E-18	2.0E-18	2.0E-18
4	8	T	2.7E-2	9.5E-4	3.5E-5	1.1E-6	2.1E-7	1.6E-7
		S	2.0E-4	2.6E-7	1.6E-7*			
	16	T	6.6E-3	5.7E-5	5.1E-7	4.4E-9	2.3E-10	1.9E-10
		S	1.2E-5	5.7E-10	1.9E-10*			
	32	T	1.6E-3	3.5E-6	7.8E-9	1.7E-11	2.3E-13	1.9E-13
		S	7.6E-7	1.7E-12	1.9E-13*			

Table 2.2.4 IDeC Methods Based on Trapezoid (T) and Simpson's (S) Rules with m = 6

			IDeC Iteration Number				
	N	Rule	0	1	2	3	4
	6	T	1.1E-2	8.1E-5	3.7E-6	4.3E-6*	
		S	2.1E-4	4.3E-6*			
Problem 2	12	T	2.7E-3	4.8E-6	4.1E-9	1.2E-8*	
		S	1.3E-5	1.2E-8*			
	24	T	6.8E-4	3.0E-7	8.4E-11	4.5E-11*	
		S	7.8E-7	4.5E-11*			
	6	T	4.9E-2	3.2E-3	2.0E-4	3.2E-5	1.7E-5
		S	6.2E-4	1.7E-5	1.8E-5*		
Problem 4	12	T	1.2E-2	1.8E-4	1.8E-6	1.2E-7	7.6E-8
		S	3.9E-5	7.3E-8	7.7E-8*		
	24	T	2.9E-3	1.1E-5	4.4E-8	4.7E-10	3.0E-10
		S	2.4E-6	2.8E-10	3.0E-10*		

Table 2.2.5 Maximum Errors in solution to Problem 5 by IDeC Method Based on Trapezoid Rule

		Iteration Number						
m	N	0	1	2	3	4	5	6
	8	2.5E-3	2.5E-4	3.1E-5	3.9E-6	4.9E-7	6.2E-8	
8	16	6.0E-4	3.5E-5	2.2E-6	1.3E-7	8.3E-9	5.2E-10	
	32	1.5E-4	4.5E-6	1.4E-7	4.3E-9	1.3E-10	4.1E-12	
	10	1.5E-3	1.3E-4	1.3E-5	1.2E-6	1.2E-7	1.2E-8	1.2E-9
10	20	3.8E-4	1.7E-5	8.7E-7	4.1E-8	2.0E-9	9.8E-11	4.8E-12
	40	9.6E-5	2.3E-6	5.4E-8	1.3E-9	3.2E-11	7.7E-13	1.9E-14
	12	1.1E-3	7.3E-5	5.9E-6	4.7E-7	3.8E-8		
12	24	2.7E-4	9.8E-6	3.9E-7	5.6E-8	6.2E-10		
	48	6.7E-5	1.3E-6	2.5E-8	5.0E-10	9.9E-12		

Table 2.2.5 gives the results for Problem 5, an example for which IDeC works well even when there is a discontinuity in f(x) of (2.2.1), $f(0)$ being defined as 1

The test results also show that the greater the degree m of the polynomials used in IDeC the higher the accuracy in its solution.

Theoretical results on the asymptotic order of the IDeC_T and IDeC_S methods for equation (2.2.1) are given in Section 3.2.2 of the next chapter.

2.3 Nonlinear Volterra Integral Equations of the Second Kind

Consider the following nonlinear Volterra integral equation (VIE) of the second kind:

$$(2.3.1) \quad y(x) - \int_0^x k(x,t,y(t))\,dt = f(x), \qquad 0 \le x \le a,$$

where k(x,t,y) is assumed to be continuous in x and t, and to satisfy a uniform Lipschitz condition in y, and f is continuous. The equation has then a unique solution which is denoted by y(x) (Davis[1962]).

An approximate numerical solution can be found by the following procedure. We choose a positive even integer N, and let $h = a/N$ and $x_r = t_r = rh$, $r = 0(1)N$. Let Y_r denote an approximation to $y(x_r)$, and f_r stand for the valu: $f(x_r)$. In (2.3.1) we set $x = rh$, and replac- the integr-l by a quadrature formula with appropriately chosen weight functions w_{rj}. Th-n we can determine Y_r from

$$(2.3.2) \quad Y_r - h \sum_{j=0}^{r} w_{rj}\, k(rh, jh, Y_j) = f_r, \qquad r = 1,2,\ldots,$$

with $Y_0 = y(0) = f_0$. In g-neral, (2.3.2) is a nonlinear equation in the single unknown Y_r and can be solv-d by an iterativ: procedur:.

We hav: consid-red thre: step-by-step proc-dures for th: nu-erical solution of th- Volterra equation (2.3.1). They ar- based on th: trapezoid rul-, Simpson's rule and mid-point rul-.

Trap-zoid rule

In th- simplest case, wh-n th- quadrature formula used is the trapezoid rule, the equations (2.3.2) have the form

$$(2.3.3) \qquad Y_r = f_r + \sum_{j=0}^{r} k(rh, jh, Y_j) - \tfrac{1}{2}k(rh, 0, Y_0) - \tfrac{1}{2}k(rh, rh, Y_r), \qquad r = 1,2,\ldots .$$

Starting with $Y_0 = f_0$, w- solv: th: first of the equations (2.3.3) for Y_1, and then solve th: second equation for Y_2 and so on.

Noble's method

For more accurate computation we use higher order quadrature formulae such as Simpson's rule. In Noble's method (Noble[1969]), we begin with two starting values Y_0 and Y_1 and compute, for

$r = 1(1)N/2$, Y_{2r} from

(2.3.4a)

$$Y_{2r} = f_{2r} + \frac{h}{3} \sum_{i=1}^{r} [k_{2r,2i-2} + 4k_{2r,2i-1} + k_{2r,2i}],$$

and Y_{2r+1} from

(2.3.4b)

$$Y_{2r+1} = f_{2r+1} + \frac{h}{3} \sum_{i=1}^{r-1} [k_{2r+1,2i-2} + 4k_{2r+1,2i-1} + k_{2r+1,2i}]$$
$$+ \frac{3h}{8}[k_{2r+1,2r-2} + 3k_{2r+1,2r-1} + 3k_{2r+1,2r} + k_{2r+1,2r+1}]$$

with $2r+1 < N$ and $k_{r,j} = k(x_r, t_j, Y_j)$.

Repeated Simpson's rule is used in (2.3.4a), whereas repeated Simpson's rule supplemented by the 3/8ths rule in (2.3.4b). The stability of the above method was proved by Noble[1969].

Mid-point rule

We may also compute the approximate values Y_i for $y(x_i)$ by the use of the mid-point rule as below.

$$Y_0 = f_0$$

$$Y_1 = f_1 + hk(x_1, x_0, Y_0)$$

$$Y_2 = f_2 + 2hk(x_2, x_1, Y_1)$$

$$\cdots\cdots\cdots\cdots\cdots$$

$$(2.3.5) \qquad Y_{2i-1} = f_{2i-1} + hk(x_{2i-1}, x_0, Y_0) + 2h \sum_{j=1}^{i-1} k(x_{2i-1}, x_{2j}, Y_{2j})$$

$$Y_{2i} = f_{2i} + 2h \sum_{j=1}^{i} k(x_{2i}, x_{2j-1}, Y_{2j-1})$$

$$\cdots\cdots\cdots\cdots\cdots$$

This is an explicit method.

2.3.1 Application of IDeC

The application of iterated defect correction to Volterra integral equations is similar to its application to Fredholm equations in Section 2.2. The details are essentially the same.

We first solve the original problem (2.3.1) by a specific method M(h), and find the discrete solution

Y_i, $i = 0(1)N$. As in Section 2.2, a smooth global approximation $P_m^{(0)}(x)$ is defined in $[0,a]$, by extending the discrete solution Y_i at x_i in $[0,a]$. We then construct the following neighbouring problem for (2.3.1):

$$(2.3.2a) \quad y(x) - \int_0^x k(x,t,y(t))\,dt = g^{(0)}(x), \qquad 0 \leq x \leq a,$$

where

$$(2.3.2b) \quad g^{(0)}(x) = P_m^{(0)}(x) - \int_0^x k(x,t,P_m^{(0)}(t))\,dt.$$

This problem is solved by the same method M(h), and the solution produced is denoted by $\eta_i^{(0)}$, $i = 0(1)N$. We then compute a better approximation to the true solution of (2.3.1) by using the equations

$$Y_i^{(1)} = Y_i + (P_m^{(0)}(x_i) - \eta_i^{(0)}), \qquad i = 0(1)N.$$

The process above may be repeated, and the solutions $Y_i^{(2)}$, $Y_i^{(3)}$, ... may be found as in Section 2.2.

Details about the implementation of the method of IDeC on quadrature methods for the numerical solution of (2.3.1) are similar to those in the case of Fredholm equation (2.2.1).

2.3.2 Numerical results

The results of some computational experiments are given in Tables 2.3.1 to 2.3.9. All the computations were done using 16-18 significant digits on the DEC-1090 computer at I.I.T. Kanpur. We programmed the three quadrature methods based on the formulae (2.3.3), (2.3.4) and (2.3.5), and the IDeC methods on these three quadrature methods. We used each method to solve the following five problems defined for x in [0,1]. Problems 1 and 3 are taken from Netravali[1973], Problem 2 and 5 from Garey[1975], and Problem 4 is from Hock[1981].

Problem 1

$$y(x) = (x - x^2) - x^4 e^{x^2} + x^2 e^{x^2} + \int_0^x x^4 e^{xt} y(t)\, dt$$

exact solution: $y(x) = x$

Problem 2

$$y(x) = e^x - x + xe^{x/2} - \int_0^x xt\, y^{\frac{1}{2}}(t)\, dt$$

exact solution: $y(x) = e^{x^2}$

Problem 3

$$y(x) = e^{-x}(1 + x + x^2) - x + \int_0^x xt\, y(t)\, dt$$

exact solution: $y(x) = e^{-x}$

Problem 4

$$y(x) = e^{-x} + \int_0^x e^{t-x}(y(t) + e^{-y(t)})\,dt$$

exact solution: $y(x) = \ln(x + e)$

Problem 5

$$y(x) = 2 - 5x + (x + 1)(3x - 1)$$

$$+ 2(x + 1)(1 - x)\log(x + 1) - 4\int_0^x \log(x - t + 1)\,y(t)\,dt$$

exact solution: $y(x) = 1 - x$.

Problem 1 to 5 were solved for $N = m$, $2m$ and $4m$ with $m = 4$, 6, 8, 10 and 12. In Tables 2.3.1 to 2.3.9, we have listed under each method the maximum values of the absolute error between the exact solution and the computed solution for different problems for the relevant values of m and N. IDeC methods based on the midpoint rule (M), trapezoid rule (T), and Noble's method (N) are denoted by IDeC_M, IDeC_T and IDeC_N respectively.

The errors listed in Tables 2.3.1 to 2.3.3 show that IDeC produces very good results in each of the three different quadrature schemes.

The relative performances of IDeC_M, IDeC_T and IDeC_N are displayed in Tables 2.3.4 and 2.3.5. The rate of increase in accuracy per iteration is rather slow for IDeC_M when compared with the rates for the other IDeCs.

Table 2.3.1 IDeC Method Based on Trapezoid Rule

			Iternation Number						
	m	N	0	1	2	3	4	5	6
Problem 1	6	6	1.5E-2	1.2E-3	1.4E-4	1.6E-5	1.8E-6	2.2E-7	2.5E-8
		12	3.5E-3	7.2E-5	2.3E-6	8.7E-8	3.6E-9	1.5E-10	1.4E-11
		24	8.7E-4	4.3E-6	3.5E-8	3.9E-10	4.5E-12	3.4E-12	3.4E-12
	8	8	8.2E-3	3.8E-4	2.6E-5	2.1E-6	1.8E-7	1.5E-8	1.3E-9
		16	2.0E-3	2.2E-5	4.1E-7	9.9E-9	2.8E-10	3.8E-12	4.6E-12
		32	4.9E-4	1.4E-6	6.3E-9	3.6E-11	3.1E-12	2.8E-12	2.8E-12
Problem 2	6	6	5.0E-3	1.7E-5	4.1E-6	4.0E-6*			
		12	1.2E-3	8.1E-7	2.6E-8	2.6E-8*			
		24	3.1E-4	4.9E-8	2.2E-10*				
	8	8	2.8E-3	4.1E-6	9.7E-8	9.2E-8*			
		16	7.0E-4	2.5E-7	2.2E-10	1.6E-10*			
		32	1.8E-4	1.6E-8	1.5E-12	1.6E-12*			
Problem 3	6	6	3.1E-3	3.1E-5	5.3E-7	1.0E-8	3.3E-9	2.9E-9*	
		12	7.7E-4	1.8E-6	6.8E-9	1.8E-11	1.4E-11*		
		24	1.9E-4	1.1E-7	1.0E-10	6.9E-13*			
	8	8	1.7E-3	9.5E-6	9.3E-8	1.5E-9	4.0E-11	8.1E-12	8.9E-12
		16	4.4E-4	5.7E-7	1.2E-9	4.2E-12	1 0E-12*		
		32	1.1E-4	3.5E-8	1.5E-11	6.4E-13	7.5E-13	7.6E 13	

			IDeC Iteration Number						
Problem	m	N	0	1	2	3	4	5	6
1	6	6	4.8E-5	5.3E-8	6.4E-9	1.1E-10	1.2E-12	8.5E-15	3.2E-17
		12	2.9E-6	3.1E-8	4.1E-11	7.7E-13	2.3E-14	3.3E-16	4.6E-17
		24	1.8E-7	1.5E-9	1.8E-11	4.2E-13	4.9E-15	1.5E-16	8.0E-17
	8	8	1.5E-5	2.3E-7	6.8E-9	9.6E-11	7.8E-13	1.2E-15	1.6E-16
		16	9.0E- 7	2.0E-8	1.5E-10	1.7E-12	5.2E-14	1.5E-15	7.5E-17
		32	5.5E- 8	9.8E-10	1.9E-11	5.0E-13	5.6E-14	6.0E-15	5.3E-16
2	6	6	5.4E-5	3.8E-6	4.0E-6*				
		12	4.2E-6	3.1E-7	2.5E-8	2.6E-8*			
		24	2.6E-7	1.1E-9	2.2E-10*				
	8	8	2.0E-5	1.5E-7	9.2E-8*	~~9.2E-8~~*			
		16	1.3E-6	1.6E-8	1.6E-10*	~~1.6E-10~~*			
		32	7.9E-8	6.5E-10	2.8E-12	1.4E-12	1.3E-12*		
3	6	6	1.4E-5	3.2E-7	2.2E-9	2.9E-9*			
		12	8.4E-7	1.0E-8	1.2E-10	1.1E-11*			
		24	5.1E-8	3.2E-10	1.0E-13	1.1E-13*			
	8	8	4.4E-6	2.0E-7	1.7E-9	7.8E-10	3.4E-12*		
		16	2.6E-7	2.6E-9	2.8E-11	9.3E-13	4.6E-14	4.1E-15	3.0E-15
		32	1.6E-8	1.5E-10	2.6E-12	1.7E-13	4.2E-15	4.6E-16	9.9E-17

Table 2.3.3 IDeC Method Based on Midpoint Rule

			Iteration Number					
	m	N	0	1	2	3	4	5
Problem 1	6	6	3.0E-2	2.6E-3	2.5E-4	2.7E-5	3.3E-6	4.1E-7
		12	8.3E-3	3.9E-4	3.8E-5	3.4E-6	3.6E-7	4.0E-8
	8	8	1.8E-2	1.5E-3	1.4E-4	9.0E-6	7.3E-7	6.1E-8
		16	4.7E-3	3.2E-4	2.9E-5	2.5E-6	2.0E-7	1.4E-8
Problem 2	6	6	1.7E-2	4.0E-4	2.5E-5	2.3E-6	4.1E-6	4.1E-6
		12	5.0E-3	2.4E-4	1.0E-6	3.4E-7	3.8E-8	2.6E-8
	8	8	1.0E-2	2.5E-4	8.7E-6	2.3E-7	9.6E-8	9.2E-8
		16	2.9E-3	1.5E-4	7.2E-6	2.8E-7	9.9E-8	1.9E-10
Problem 3	6	6	7.5E-3	7.2E-4	5.8E-5	5.7E-6	4.8E-7	3.3E-8
		12	2.0E-3	4.3E-4	4.7E-5	4.1E-6	3.2E-7	2.4E-8
	8	8	4.4E-3	4.2E-4	3.6E-5	2.4E-5	1.4E-6	7.5E-8
		16	1.1E-3	3.2E-4	3.5E-5	3.0E-6	2.5E-7	1.9E-8

Table 2.3.4 Problem 4: IDeC methods based on formulae M. T, N with m = N

N	Basic Method	IDeC Iteration number						
		0	1	2	3	4	5	6
	M	1.7E-2	1.4E-3	1.6E-4	1.4E-5	9.6E-7	5.7E-8	
6	T	3.7E-3	4.2E-5	1.5E-6	6.8E-8	3.0E-9	7.4E-10	7.2E-10
	N	1.2E-5	3.6E-7	3.1E-9	2.4E-10	7.2E-10*		
	M	1.0E-2	1.4E-3	1.8E-4	1.5E-5	6.3E-7	1.5E-7	
8	T	2.1E-3	1.5E-5	4.8E-7	2.0E-8	7.4E-10	2.5E-11	4.1E-12
	N	3.6E-6	2.9E-7	2.5E-9	1.5E-11	4.7E-12	4.3E-12*	

Table 2.3.5 Problem 5: Results for $IDeC_M$, $IDeC_T$ and $IDeC_N$ with m = 6

		Iteration number						
N	Basic Method	0	1	2	3	4	5	6
	M	6.1E-2	2.5E-2	1.1E-3	4.0E-4	2.9E-5	5.3E-6	
6	T	6.7E-3	1.4E-4	3.6E-6	8.7E-8	2.1E-9	5.3E-11	1.3E-12
	N	5.4E-5	3.6E-6	4.8E-8	3.4E-9	4.3E-11	3.7E-12	5.3E-14
	M	2.0E-2	1.4E-2	2.4E-3	1.7E-4	5.6E-6	1.2E-7	
12	T	1.6E-3	9.3E-6	5.8E-8	3.8E-10	2.3E-12	1.8E-14	1.5E-16
	N	2.6E-6	1.6E 7	1.5E-9	1.8E-11	3.6E-13	5.9E-15	8.7E-18

Table 2.3.6 IDeC$_M$ method with $m = 4$ and $h = 1/m$

	Iteration Number				
Problem	0	1	2	3	4
2	3.1E-2	7.5E-4	2.2E-4	1.6E-4	1.6E-4
3	1.6E-2	1.8E-3	2.5E-4	2.5E-5	7.5E-7
4	3.6E-2	1.9E-3	1.1E-4	1.0E-5	8.6E-7

Table 2.3.7 IDeC methods using polynomials of degree $m = N = 12$

	IDeC$_r$ iteration			IDeC$_N$ iteration		
Problem	2	4	6	2	4	6
1	2.4E-6	5.0E-9	2.4E-11	4.4E-9	5.8E-13	9.5E-17
2	5.2E-10	3.3E-11	3.3E-11	4.6E-10	3.4E-11	3.4E-11
3	8.0E-9	5.3E-12	4.2E-12	1.9E-9	1.9E-14	4.5E-17
4	1.1E-7	1.3E-10	2.1E-12	1.9E-9	4.4E-12	2.1E-15
5	2.4E-7	4.6E-11	8.8E-15	7.6E-9	3.9E-11	7.0E-14

Table 2.3.8 Modified increment methods, and the IDeC$_T$ method with m = 10

Problem	N	Method A	Method B	IDeC$_T$ interation number 1	2	4	6
2	10	1.9E-7	2.0E-7	1.6E-6	3.3E-9	1.9E-9	1.9E-9
	20	1.6E-8	1.7E-8	1.0E-7	2.3E-11	1.8E-12	1.9E-12
5	10	1.6E-7	6.2E-8	2.9E-5	5.1E-7	1.4E-10	3.7E-14
	20	8.4E-9	7.0E-9	1.9E-6	8.2E-9	1.7E-12	6.1E-14

Table 2.3.9 Spline method of Netravali, and the IDeC$_T$ method with m = 10

Problem	N	Netravali	IDeC$_T$ iteration number 1	2	4	6
1	10	8.8E-5	1.5E-4	7.1E-6	2.6E-8	1.2E-10
	20	8.8E-7	9.1E-6	1.1E-7	3.4E-11	1.7E-12
2	10	1.0E-5	3.8E-6	2.4E-8	4.6E-12	3.5E-12
	20	8.7E-8	2.3E-7	3.0E-10	1.3E-12	1.3E-12

There is not a marked difference between the performances of $IDeC_T$ and $IDeC_N$, except that to obtain a given accuracy the number of iterations required by $IDeC_N$ is smaller than the number required by $IDeC_T$.

Table 2.3.6 illustrates the efficiency of the $IDeC_M$ method for a relatively large step size, and Table 2.3.7 that of the $IDeC_T$ and $IDeC_N$ methods with the use of a relatively large degree of polynomials.

The results obtained by the $IDeC_T$ method based on the trapezoid rule are contrasted with those by the modified increment methods A and B of Garey[1975] in Table 2.3.8, and with those by the spline approximation method of Netravali[1973] in Table 2.3.9. The $IDeC_T$ method produces much more accurate results than any of the basic discretization methods even for relatively small values of N.

We conclude this section with the observation that the performance of IDeC methods based on quadrature schemes for non-linear VIEs is indeed very good in practice.

2.4 Linear Volterra Integral Equations of the Second Kind

We now consider the numerical solution of the following Volterra integral equation of the second kind:

$$(2.4.1)\quad y(x) - \int_0^x k(x,t)\, y(t)\, dt = f(x), \qquad 0 \le x \le a.$$

We assume that $k(x,t)$ is continuous for $0 \leq x,t \leq a$ and that $f(x)$ is continuous for $0 \leq x \leq a$. Then a continuous solution $y(x)$ exists (Baker[1977]).

We choose a positive even integer N and let $h = a/N$. For $r = 0(1)N$, we substitute $x = rh$ in (2.4.1) and use a quadrature formula to approximate the integral

$$\int_0^{rh} k(rh,t)\, y(t)\, dt.$$

This leads to the equations

$$(2.4.2) \quad Y(rh) - \sum_{j=0}^{r} w_{r,j}\, k(rh,jh)\, Y(jh) = f(rh), \quad r = 1(1)N,$$

where $Y(rh)$ denotes the approximation to $y(rh)$ and the $w_{r,j}$ are the weights associated with the quadrature formula used.

Thus, starting with the initial approximation $Y(0) = y(0) = f(0)$, we obtain the equations

$$- w_{1,0}\, k(h,0)\, Y(0) + [1 - w_{1,1}\, k(h,h)]\, Y(h) = f(h),$$

$$(2.4.3) \qquad \cdots\cdots\cdots\cdots$$

$$\cdots\cdots\cdots\cdots$$

$$- \sum_{j=0}^{N-1} w_{N,j}\, k(Nh,jh)\, Y(jh)$$

$$+ [1 - w_{N,N}\, k(Nh,Nh)]\, Y(Nh) = f(Nh)$$

From the above successive equations, the values Y(h), Y(2h), ..., Y(Nh) are calculable one after another. We will call this a quadrature method, since we have adapted here the idea of the quadrature method for Fredholm equations. More details on (2.4.1) are given in Baker[1977].

Application of IDeC and Numerical results

The details regarding the application of IDeC on the quadrature method based on the trapezoid rule for the linear VIE (2.4.1) in Section 2.3.1, except that now the sth neighbouring problem for (2.4.1) is of the form:

$$(2.4.4) \qquad y(x) - \int_0^x k(x,t)\, y(t)\, dt$$
$$= r_m^{(s)}(x) - \int_0^x k(x,t)\, p_m^{(s)}(t)\, dt, \qquad 0 \le x \le a.$$

We implemented the quadrature and IDeC methods applicable to (2.4.1). We give here results from the computational tests on the following sample problem taken from Baker[1977]:

Problem

$$y(x) - 2\int_0^x \cos(x - t)\, y(t)\, dt = e^x$$

xact solution: $y(x) = e^x(1 + x)^2$.

More results on a few other sample problems are included in Chapter 8. The calculations were performed on the ICL 1904S computer at Banaras Hindu University, Varanasi in single precision (9-11 decimals) arithmetic.

We choose the number N of intervals in [0,1] to be 4(4)16, and the degree m of the polynomial elements of the function $P_m^{(s)}(x)$ in (2.4.4) to be N. Maximum absolute errors in the computed solution by the quadrature and IDeC methods are given in Table 2.4.1 below.

The first iterate of the IDeC method based on the trapezoid rule is as good as the result obtained by the method which uses the weights of repeated Simpson's rule with the 3/8ths rule. The latter method gives the errors 3×10^{-3} and 2×10^{-4} for N = 8 and N = 16 respectively (see page 785 of Baker[1977]).

Table 2.4.1 The quadrature and IDeC methods based on trapezoid rule

N	Quadrature method	IDeC method iterations 1	2	3	4	5
4	4.4E-1	3.8E-2	3.6E-3	9.9E-4	7.6E-4	7.9E-5
8	1.1E-1	3.0E-3	2.0E-4	2.1E-5	2.4E-6	2.7E-7
12	5.1E-2	7.1E-4	4.6E-5	4.8E-6	5.1E-7	5.5E-8
16	2.9E-2	2.5E-4	5.5E-6	4.1E-6	1.2E-6	4.3E-7

The performance of IDeC on linear Volterra integral equation (2.4.1) is quite good, even for relatively large values of h.

2.5 Eigenvalue Problems

In this section we consider the numerical solution of the Fredholm integral equation of the third kind given by

$$(2.5.1)\qquad \int_a^b k(x,t)\ y(t)\ dt = \lambda y(x), \qquad a \leq x \leq b,$$

where the kernel $k(x,t)$ is given, and the eigenvalues λ and the corresponding eigenfunctions $y(x)$ are to be calculated.

As in Section 2.2, a simple numerical method for (2.5.1) is to replace the integral by a quadrature rule, and collocate at the points $x = x_i$. This leads to

$$(2.5.2)\qquad \sum_{j=0}^{N} w_j\ k(x_i,t_j)Y_j = \Lambda\ Y_i, \qquad i = 0(1)N,$$

where Y_i and Λ stand for an approximation to $y(x_i)$ and λ respectively. In matrix notation, this becomes the matrix eigenvalue problem

$$(2.5.3)\qquad K\bar{Y} = \Lambda\bar{Y}$$

where $K = [w_j\ k(x_i,t_j)]$ and $\bar{Y} = Y_0,Y_1,\ \ldots,\ Y_N)^T$.

Solving (2.5.3) we obtain N+1 eigenvalues Λ_r, $r = 0(1)N$ and the corresponding eigenvectors $\bar{Y}_r$. The above method is the 'quadrature method' for the eigenvalue problem (2.5.1) (Baker[1977]).

A method for iterative improvement

Since the application of the method of IDeC to eigenvalue problems is not straightforward, we shall now describe a modified method for estimating the errors and improving the results for the eigenvalue problem (2.5.1).

Suppose the quadrature method based on the quadrature rule R produces an eigenvalue Λ_0 of the integral, equation (2.5.1), and the corresponding eigenvector $\bar{Y}_0^{(o)}$ which approximates the required eigenfunction. Let the function $p^{(o)}(x)$ be a smooth global approximation to the eigenfunction corresponding to the eigenvalue Λ_0 of (2.5.1). This function is computed by fitting a polynomial to the data points $(x_i, Y_{o,i}^{(o)})$ in a least square sense. Then consider the problem

$$(2.5.4a)\quad \int_a^b k(x,t)\, y(t)\, dt = \Lambda_0 y(x) + g^{(o)}(x),$$

where

$$(2.5.4b)\quad g^{(o)}(x) = \int_a^b k(x,t)\, p_0^{(o)}(t)\, dt - \Lambda_0 p_0^{(o)}(x).$$

The exact solution for the above problem is $P_0^{(o)}(x)$. We notice that (2.5.4a) is <u>not</u> an eigenvalue problem of the form (2.5.1), but is a linear Fredholm equation of the form (2.2.1).

We solve (2.5.4) by the corresponding quadrature method based on the quadrature rule R, and denote its solution by $\bar{\eta}_0^{(o)}$. Adapting the idea of Zadunaisky, we expect $\bar{Y}_0^{(1)}$ given by

$$\bar{Y}_0^{(1)} = \bar{Y}^{(o)} + (\bar{P}_0^{(o)} - \bar{\eta}_0^{(o)}) \tag{2.5.5}$$

to be a better approximation to the eigenfunction than $\bar{Y}^{(o)}$. The use of (2.5.3) will then give an improved eigenvalue $\bar{\Lambda}_0^{(1)}$.

This process may be repeated for the determination f of $(\bar{Y}_0^{(2)}, \bar{\Lambda}_0^{(2)}, (\bar{Y}_0^{(3)}, \bar{\Lambda}_0^{(3)}), \ldots$.

One difficulty in this method is that the integral equation (2.5.4a) is ill-conditioned, because of the eigenvalue Λ_0 of K. (See Noble[1977], p 919).

2.6 Conclusion

We have successfully applied the method of iterated defect correction on the quadrature methods for linear Fredholm, and linear and nonlinear Volterra integral equations of the second kind.

In the case of Fredholm equations, our computational experiments show that the greater the order of the basic quadrature method, the faster the convergence of the IDeC method, and that an increase in the degree m of the polynomials yields an increase in the accuracy of the IDeC solution. The properties exhibited by our numerical results pave the way for our theoretical analysis of the IDeC method presented in the next chapter.

As for the application to nonlinear Volterra integral equations, we have considered IDeC methods based on quadrature methods which use the mid-point rule and the trapezoid rule and the IDeC method based on Noble's method. If we arrange these three IDeC methods in the increasing order of accuracy in their solutions, they will appear in the sequence mentioned above. However, the IDeC method which uses the trapezoid rule is the most preferable of the three methods since its underlying quadrature method is self-starting. It is remarkable that the IDeC methods for a given step-size generally produce more accurate solutions than any other discretization methods with the same step-size.

Again in the case of linear VIEs, the IDeC method based on the quadrature method that uses the trapezoid rule yields highly accurate solutions.

However, application of the method of IDeC to eigenvalue problems is not straightforward and presents difficulties in contrast to Richardson extrapolation and deferred correction techniques which are easily applicable.

On the whole, IDeC methods are easy to implement and very effective in practice. Moreover, they produce highly accurate solutions to problems in Fredholm and Volterra integral equations of the second kind.

CHAPTER 3

THE METHOD OF SUCCESSIVE EXTRAPOLATED ITERATED DEFECT CORRECTION

3 1 Introduction

In Section 2.2 of the last chapter we described an application of iterated defect correction on the quadrature methods for the numerical solution of Fredholm's integral equations of the second kind. In this chapter we derive an asymptotic expansion in even powers of stepsize h for the global error in the solution produced by the IDeC method, and show how the numerical results of Section 2.2.2 confirm our theoretical analysis.

Applying Richardson extrapolation repeatedly on the IDeC method, we present in Section 3.3 the technique of Successive Extrapolated Iterated Defect Correction (SEIDeC) and the resulting asymptotic expansion for the global error. Numerical examples are given in Section 3.4 to demonstrate the superiority of SEIDeC over IDeC.

The implementation of SEIDeC is easy and gives a very fast convergence for smooth kernels and functions.

3.2 Asymptotic Expansion for IDeC

An application of the IDeC principle on the quadrature methods for the second kind, linear Fredholm integral equations (2.2.1) was discussed in Section 2.2 of the previous chapter. Now an asymptotic expansion for the global error in the solution produced by the IDeC method is derived in this section.

We let $g(x_i) = g_i$ and denote the column vector $(g_0, g_1, \ldots, g_N)^T$ by $\bar{g}$. We can then represent the linear system of equations (2.2.2)

$$Y_i - \sum_{j=0}^{N} w_j \; k(x_i, t_j) Y_j = f(x_i), \qquad i = 0(1)N$$

in matrix notation as

$$(3.2.1) \qquad (I - K)\bar{Y} = \bar{f}$$

where I and K are $(N + 1) \times (N + 1)$ matrices. But the vector $\bar{Y} = (y(x_0), \ldots, y(x_N))^T$ satisfies

$$(3.2.2) \qquad (I - K)\bar{Y} = \bar{f} + \bar{\tau}$$

where the error $\bar{\tau}$ in the quadrature formula is given by

$$(3.2.3) \quad \tau_i = \int_a^b k(x_i, t) \; y(t) \; dt - \sum_{j=0}^{N} w_j k(x_i, t_j) \; y(t_j).$$

Substracting (3.2.1) from (3.2.2), we have

$$(I - K)(\bar{y} - \bar{Y}) = \bar{\tau}$$

Now if the non-singularity of (I - K) is assumed, the error in the computed solution $\bar{Y}$ can be written as

$$(3.2.4) \qquad \bar{y} - \bar{Y} = (I - K)^{-1}\,\bar{\tau}.$$

Similarly, for any fixed r = 0, 1, 2, ... the error in the computed solution

$$\bar{\eta}^{(r)} = (\eta_0^{(r)}, \eta_1^{(r)}, \ldots, \eta_N^{(r)})^T$$

for NP (2.2.3) or (2.2.5) is given by

$$(3.2.5) \qquad \bar{P}_m^{(r)} - \bar{\eta}^{(r)} = (I - K)^{-1}\,\bar{\varepsilon}^{(r)}, \text{ where}$$

$$(3.2.6) \quad \varepsilon_i^{(r)} = \int_a^b k(x_i, t)\, P_m^{(r)}(t)\, dt - \sum_{j=0}^{N} w_j k(x_i, t_j) P_m^{(r)}(t_j)$$

Now for the quadrature method $\text{Quad}_T(h)$ based on the trapezoid rule, the weights w_j in (2.2.2) are chosen such that

$$w_0 = w_N = h/2, \qquad w_j = h \text{ for } j = 1(1)N-1,$$

and the quadrature error τ_i in (3.2.3) takes the form (see Mayers[1974] and Walsh[1974])

$$(3.2.7) \quad \tau_i = A_2(x_i)h^2 + A_4(x_i)h^4 + A_6(x_i)h^6 + \ldots..$$

We assume that (I - K) is <u>stable</u> in the sense of Noble (see Noble[1977], pp. 921-922). Then because of equations (3.2.4) and (3.2.7), the order of the error $Y_i - y(x_i)$ in the approximate solution Y_i is the same as the order of the quadrature error, and we can write

$$(3.2.8) \quad Y_i - y(x_i) = B_2(x_i)h^2 + B_4(x_i)h^4 + B_6(x_i)h^6 + \dots,$$

Let both $Y(x)$ and $Y(x,h)$ represent a continuous approximation to $y(x)$ of (2.2.1) constructed from the discrete solution Y_i obtained by the method $\mathrm{Quad}_T(h)$. Then, because of (3.2.8), we have

$$(3.2.9) \quad Y(x,h) - y(x) = B_2(x)h^2 + B_4(x)h^4 + B_6(x)h^6 + \dots,$$

where $B_2(x)$, $B_4(x)$, etc. are suitable functions.

Now, for each j, the expression $P_{m,j}^{(0)}(x) - Y(x)$ is the error in the polynomial $P_{m,j}^{(0)}(x)$ due to interpolation (see Ralston and Rabinowitz[1978]) of the points (x_i, Y_i) with $x_i \in I_j$ (the meaning of I_j as in Section 2.2.1). Since $P_m(x) = (P_m(x) - Y(x)) + Y(x)$, we have from (3.2.9),

$$(3.2.10) \qquad P_m^{(0)}(x) = y(x) + B_2(x)h^2 + O(h^4).$$

Consequently, for the NP (2.2.3), the equation (3.2.6) yields

$$\varepsilon_i^{(0)} = \tau_i + h^2\Big[\int_a^b k(x_i,t)B_2(t)\,dt - \sum_{j=0}^{N} w_j k(x_i,t_j)B_2(t_j)\Big] + O(h^4)$$

$$= \tau_i + O(h^4), \text{ with the aid of (3.2.7).}$$

We can therefore write $\varepsilon_1^{(0)}$ as

$$(3.2.11) \quad \varepsilon_1^{(0)} = A_2(x_1)h^2 + \alpha_4(x_1)h^4 + \alpha_6(x_1)h^6 + \ldots\ldots$$

Moreover, from (2.2.1), (2.2.3b) and (3.2.10) we have

$$g_m^{(0)}(x_1) - f(x_1) = O(h^2).$$

It follows from (3.2.5) that

$$(3.2.12) \quad \eta_1^{(0)} - P_m^{(0)}(x_1) = B_2(x_1)h^2 + \beta_4(x_1)h^4 + \beta_6(x_1)h^6 + \ldots$$

which is analogous to the expansion (3.2.8) for $Y_1 - y(x_1)$.

The use of Zadunaisky's principle

$$Y_1^{(1)} = Y_1 + P_m^{(0)}(x_1) - \eta_1^{(0)}$$

yields the equation

$$(3.2.13) \quad Y_1^{(1)} - y(x_1) = B_4^{(1)}(x_i)h^4 + B_6^{(1)}(x_1)h^6 + \ldots,$$

$$\text{with } B_j^{(1)} = B_j - \beta_j, \qquad j = 4,\ 6,\ \ldots$$

In a similar manner we can show that the following result holds, assuming that both $Y^{(r)}(x,h)$ and $Y(x)$ denote the approximation to $y(x)$ of (2.2.1) produced after the rth iteration of IDeC.

For $r = 1, 2, \ldots,$ we have

$$(3.2.14) \quad Y^{(r)}(x,h) - y(x) = B_q^{(r)}(x)h^q + B_{q+2}^{(r)}(x)h^{q+2} + \ldots,$$

and

$$(3.2.15) \qquad g_m^{(r)}(x_i) - f(x_i) = O(h^q),$$

where $q = \min(2r + 2,\ m + 2)$; since for any fixed r, we have

$$P_m^{(r)}(t_j) = Y^{(r)}(t_j), \qquad j = 0(1)N,$$

and therefore, (see Ralston and Rabinowitz[1978])

$$\int_a^b k(x_i,t)(P_m^{(r)}(t) - Y^{(r)}(t))\,dt = O(h^{m+2}).$$

Thus in the case of the trapezoid rule, each iteration r, with $1 \leq r \leq m/2$, of the IDeC increases the order of accuracy in the solution by two, and the asymptotic expansion for the global error in the final solution $Y_{IDeC}(x,h)$ of the IDeC method is given by

$$(3.2.16) \quad Y_{IDeC}(x,h) - y(x) = C_{m+2}(x)h^{m+2} + C_{m+4}(x)h^{m+4} + \dots$$

If the quadrature method Quad(h) is based on Simpson's rule instead of the trapezoid rule, the term containing h^2 does not appear in equations (3.2.7) to (3.2.10), and the relation between the errors in the approximate solution of the original problem and the first neighbouring problem encountered in the defect correction procedure is given by

$$(3.2.17) \qquad \varepsilon_1^{(0)} = \tau_1 + O(h^8).$$

Consequently, we can show that

$$(3.2.18) \quad Y_1^{(1)} - y(x_1) = B_8^{(1)}(x_1)h^8 + B_{10}^{(1)}(x_1)h^{10} + \dots$$

and that the resulting asymptotic expansion for the IDeC method remains the same as (3.2.18).

In fact, for a fixed m, the asymptotic expansion (3.2.16) for the global error in the solution of (2.2.1) is independent of the underlying Newton-Cotes formula which the IDeC method uses. Thus the error is of order h^{m+2}, where h is the stepsize and m is the degree of the polynomial employed in the neighbouring problems.

Thus we have the following theorem:

Theorem 3.2.1

If the order n of the quadrature method for (2.2.1) is 2 or 4, and the degree m of interpolating polynomials is even, then the solution $Y_1^{(r)}$ of the rth iterate of IDeC satisfies

$$(3.2.19) \quad Y_1^{(r)} - y(x_1) = O(h^{\min(n + rn,\ m + 2)}), \quad r = 0,\ 1,\ 2,\ \dots,\ \frac{m+2}{n}-1$$

Moreover, the asymptotic expansion for the global error in the final solution of $Y_{IDeC}(x,h)$ of the IDeC has the form (3.2.16).

Numerical results

For the results on the order of convergence for the IDeC method applied to the test-problems in Fredholm equations, we refer to the tables in Section 2.2.2, and compute the ratios of the actual computed errors in the solution by the IDeC(i,h) and IDeC(i, $\frac{1}{2}$h) methods. These ratios denoted by V(i,h) are defined as follows:

$$(3.2.20) \qquad V(i,h) = \frac{\text{error by IDeC}(i,h)}{\text{error by IDeC}(i,\frac{h}{2})} ,$$

where i is the iteration number and h the stepsize. From these values the orders of IDeC are calculated for a few examples. Table 3.2.1 lists these values and the orders of IDeCs based on the trapezoid rule, and Table 3.2.2 those based on Simpson's rule. In both the tables L denotes the fact that the machine accuracy has been reached for at least one of the IDeC solution values from which V(i.h) has been calculated

The results in Tables 3.2.1 and 3.2.2 on the orders of convergence for the IDeC iterates are in accordance with the theoretical results of Theorem 3.2.1.

Table 3.2.1 The values $V(i,h)$ of (3.2.20) and the orders for the ith iteration of $IDeC_T$.

			Iteration Number						
n	Problem	h	0	1	2	3	4	5	6
8	1	1/8	4	16	65	253	997	997	
		1/16	4	16	64	255	1009	942	
		Order	2	4	6	8	10	10	
10	4	1/10	4	17	67	273	916	3296	3207
		1/20	4	16	65	260	978	4120	4134
		Order	2	4	6	8	10	12	12
12	2	1/12	4	16	64	231	16566		
		1/24	4	16	64	257	L		
		Order	2	4	6	8	14		

Table 3.2.2 The values V(i,h) and the orders for the ith iteration of $IDeC_S$.

				Iteration Number		
m	Problem	h	0	1	2	3
10	4	1/10	16	236	3217	3213
		1/20	16	254	4156	4164
		Order	4	8	12	12
12	3	1/12	16	250	4023	L
		1/24	16	254	L	L
		Order	4	8	12	

However, in the case of Problem 3, the computed orders for $IDeC_T$ with m = 8 were found to be 2,4,6,8,10, and 12, and in case of Problem 5 they were 2,3,4,5,6,7 and 8 (refer Section 2.2.2). The extra order of convergence in the former case is attributed to the fact that since the exact solution is y(x) = x, the polynomials $P_m^{(r}$ which extend the discrete solution do not introduce an error due to interpolation. In the latter case, the increase in the order of convergence is not two, because of the discontinuity in the function f(x).

3.3 Successive Extrapolated Iterated Defect Correction (SEIDeC)

3.3.1 Asymptotic expansion for SEIDeC

Now that there exists an asymptotic expansion (3.2.16) for the IDeC method on the quadrature methods for the linear Fredholm equation (2.2.1), Richardson extrapolation or 'deferred approach to the limit' (see Ralston and Rabinowitz[1978]) may be used as explained below.

By computing the approximation $Y_{IDeC}(x,h)$ by the IDeC method for different values of h, and combining two approximations which have similar errors, we can eliminate successive error terms C_{m+i}, $i = 2, 4, \ldots$ in (3.2.16), and thus obtain more and more accurate approximations.

For instance, suppose we obtain an approximation $Y_{IDeC}(x,h)$ with a particular stepsize h, and then halve the stepsize and get a second approximation $Y_{IDeC}(x, \frac{1}{2}h)$, and finally find a new approximation $Y^1_{IDeC}(x,h)$ by combining the first two approximations as in the equation

$$(3.3.1) \qquad Y^1_{IDeC}(x,h) = \frac{2^{m+2}Y_{IDeC}(x,\frac{1}{2}h) - Y_{IDeC}(x,h)}{2^{m+2} - 1}.$$

Then we obtain

$$(3.3.2) \qquad Y^1_{IDeC}(x,h) = y(x) + C^1_{m+4}(x)h^{m+4} + O(h^{m+6}),$$

which has no h^{m+2} term.

If we now determine a third approximation, again by halving the stepsize $\frac{1}{2}h$, we can combine it with the second to obtain another new approximation $Y^{1}_{IDeC}(x,\frac{1}{2}h)$ containing an error expressible as $C^{1}_{m+4}(x)h^{m+4} + O(h^{m+6})$. From the new values $Y^{1}_{IDeC}(x,h)$ and $Y^{1}_{IDeC}(x,\frac{1}{2}h)$, we can combine a still another new approximation $Y^{2}_{IDeC}(x,h)$, from an equation similar to (3.3.1) but with the exponent m+4 instead of m+2. Now the expansion for the approximation $Y^{2}_{IDeC}(x,h)$ has no $C^{1}_{m+4}(x)h^{m+4}$ term.

In general, after the elimination of the first j coefficients in (3.2.16), the form of the asymptotic expansion is

$$(3.3.3) \qquad Y^{j}_{IDeC}(x,h) = y(x) + C^{j}_{m+2+2j}(x)h^{m+2+2j} + C^{j}_{m+4+2j}(x)h^{m+4+2j} + \ldots..$$

Thus the principle of Richardson extrapolation may be performed successively, using the solutions produced by the IDeC method for various stepsizes h. We call this technique the method of Successive Extrapolated Iterated Defect Correction (SEIDeC).

3.3.2 Implementation

The method SEIDeC is convenient and efficient for automatic computation of the numerical solution of (2.2.1), provided the unknown solution $y(t)$ and the kernel $k(x,t)$ are sufficiently smooth.

Calling the process of computing the values $Y^{j}_{IDeC}(x,h)$ as the _jth order extrapolation_, we illustrate the procedure for computing the SEIDeC solution for (2.2.1) upto an extrapolation of a given order j_{max} or until the value from the latest extrapolation completed differs from the most accurate value of previous extrapolation by less than a tolerance TOL.

Begin by calculating the IDeC solution values for an appropriate stepsize h and then for $\frac{h}{2}$, and entering them along with the solution from the first order extrapolation as in the following table:

Stepsize	h		$\frac{1}{2}h$
Solution	$Y_{IDeC}(x,h)$		$Y_{IDeC}(x,\frac{h}{2})$
1st order extrapolation		$Y^{1}_{IDeC}(x,h)$	

If the condition $\left|Y^{1}_{IDeC}(x,h) - Y_{IDeC}(x,\tfrac{1}{2}h)\right| < TOL$ is satisfied or if $j_{max} = 1$, then take $Y^{1}_{IDeC}(x,h)$ as the required value and terminate the procedure. Otherwise, find the IDeC solution with stepsize $\frac{h}{4}$, and expand the table to the following:

Stepsize	h	$\frac{1}{2}h$	$\frac{1}{4}h$
Solution	$Y_{IDeC}(x\ h)$	$Y_{IDeC}(x,\frac{1}{2}h)$	$Y_{IDeC}(x,\frac{1}{4}h)$
1st order extrapolation	$Y^1_{IDeC}(x,h)$	$Y^1_{IDeC}(x,\frac{1}{2}h)$	
2nd order extrapolation		$Y^2_{IDeC}(x,h)$	

If either of the two conditions

$$\left| Y^2_{IDeC}(x,h) - Y^1_{IDeC}(x,\tfrac{h}{2}) \right| < \text{TOL} \quad \text{or} \quad j_{max} = 2 \text{ is satisfied,}$$

take $Y^2_{IDeC}(x,h)$ as the output and terminate the procedure. Otherwise find the IDeC solution with stepsize $\frac{h}{8}$, and expand the above table again.

In the computer memory this table is represented as a two-dimensional array of the form

$$\begin{array}{llll} A_{00} & A_{01} & A_{02} & A_{03} \\ A_{10} & A_{11} & A_{12} & \\ A_{20} & A_{21} & & \\ A_{30} & & & \end{array}$$

where $A_{ij} = Y^i_{IDeC}(x,\frac{h}{2^j})$.

3.4 Numerical Results

In this section we give computational results on the application of SEIDeC on quadrature methods for Fredholm integral equations of the second kind. We consider here

Problems 1 and 2 of Section 2.2.2.

The degree m of the polynomials used was taken to be 4 and 6. Maximum absolute errors in the final solution produced by the IDeC and SEIDeC methods with m = 4 and m = 6 are given in Tables 3.4.1 and 3.4.2 respectively. In these tables the SEIDeC method with the ith order extrapolation is denoted by SEIDeC(i).

Table 3.4.1 IDeC and SEIDeC methods with m = 4

	Problem 1			Problem 2		
Method N	12	24	48	12	24	48
IDeC	1.2E-8	1.9E-10	2.9E-12	3.4E-7	5.1E-9	8.0E-11
SEIDeC(1)	6.8E-13	2.7E-15		1.4E-10	5.5E-13	
SEIDeC(2)	9.5E-18			1.6E-14		

Table 3.4.2 IDeC and SEIDeC methods with m = 6

	Problem 1			Problem 2		
Method N	12	24	48	12	24	48
IDeC	5.5E-11	2.1E-13	8.2E-16	1.2E-8	4.5E-11	1.7E-13
SEIDeC(1)	1.7E-15	1.7E-18		3.1E-12	2.9E-15	
SEIDeC(2)	0.0E-00			1.9E-16		

We find that the computed order of SEIDeC(1) is 8 when m = 4, and 10 when m = 6. This is in accordance with our theoretical result (3.3.3).

In practical situations the method of SEIDeC is very efficient.

3.5 Conclusion

We have studied the asymptotic expansion for the global errors associated with the IDeC technique applied to quadrature methods for FIEs. We have shown that the order of convergence of an IDeC iteration is equal to the order of the quadrature method on which it is based. This is in accordance with the general result for IDeC proved by Stetter[1978].

Our study of IDeC has permitted the application of Richardson extrapolation on the IDeC method and the presentation of the new acceleration technique SEIDeC together with its asymptotic expansion. As a consequence, high order accuracy in the numerical solution of FIE (2.2.1) has been obtained with only a modest amount of work. Another feature of SEIDeC is that its order of convergence is unlimited, whereas that of IDeC is limited by the degree of the polynomials used in the defect correction procedure.

We have also described how SEIDeC is easily implemented as an automatic procedure. Numerical examples demonstrate the efficiency and the fast convergence of the new method.

An application of SEIDeC is feasible only when the underlying IDeC method admits an asymptotic expansion in powers of stepsize h. In the case of Volterra integral equations, it is not easy to discover whether such an asymptotic expansion exists for the IDeC technique based on a given quadrature method. Moreover, our numerical results for these equations in Sections 2.3 and 2.4 do not exhibit any such asymptotic behaviour. Consequently, we have not considered SEIDeC methods for Volterra integral equations.

CHAPTER 4

THE METHODS OF ITERATED DEFERRED DEFECT CORRECTION

4.1 Introduction

In this chapter we present a new method of iterative improvement, which is the result of applying the principle of Iterated Defect Correction (IDeC) on the method of Iterated Deferred Correction (IDC). We call this new method Iterated Deferred Defect Correction (IDDeC). The method IDDeC is a generalization of the methods IDC and IDeC. It preserves the good features of its constituent methods, and is free from their disadvantages. Our method, like its constituents, finds solutions of increasingly high accuracy on the same original grid (in contrast to the Richardson extrapolation process which requires solutions also on finer grids).

Section 4.1.1 contains a description of the method of iterated deferred correction as applied to FIE (2.2.1), and Section 4.2 a derivation of the asymptotic expansion of the method.

In Section 4.3 we describe the method of IDDeC, and analyze its asymptotic behaviour. In Section 4.4 we tabulate the numerical results obtained when IDDeC was applied to linear FIEs and to linear and nonlinear VIEs. Our implementation of the IDDeC and IDC methods were based on the trapezoid rule with Gregory correction and/or Simpson's rule with correction.

4.1.1 Iterated Deferred Correction (IDC)

In order to explain the idea of deferred correction, we use the linear Fredholm integral equation (2.2.1) of the second kind:

$$(4.1.1) \qquad y(x) - \int_a^b k(x,t)\, y(t)\, dt = f(x), \qquad a \leq x \leq b.$$

We assume that the functions $k(x,t)$ and $f(x)$ possess continuous derivatives of sufficiently high order, so that the unique solution $y(x)$ exists and also has derivatives of sufficiently high order.

To introduce the discretization of (4.1.1), we let N be a positive integer, $h = (b-a)/N$, and $x_i = t_i = a + ih$, $i = 0(1)N$. Furthermore, we denote $g(x_i)$ by g_i, where $g(x)$ is any function on $[a,b]$, and represent the column vector $(g_0, g_1, \ldots, g_N)^T$ by $\bar{g}$.

We then consider the Gregory's formula

$$(4.1.2) \quad \int_a^b u(t)\, dt = h(\tfrac{1}{2}u_0 + u_1 + \ldots + u_{N-1} + \tfrac{1}{2}u_N)$$
$$+ h \sum_{s \geq 1} c_s(\nabla^s u_N + (-1)^s \Delta^s u_0).$$

with $c_1 = -1/12$, $c_2 = -1/24$, $c_3 = -19/720$, $c_4 = -3/160$, etc.

(Baker[1977]). We first neglect the second term on the right hand side of (4.1.2) and use the resulting trapezoid rule to write (4.1.1) as

$$(4.1.3)\quad Y(x) - h \sum_{j=0}^{N} w_j\, k(x\,,t_j)\, Y(t_j) = f(x), \quad a \leq x \leq b,$$

where $Y(x)$ is an approximation to $y(x)$, and the w_j are the weights: $w_0 = w_N = \frac{1}{2}$ and $w_j = 1$ for $j = 1(1)N-1$. The substitution $x = x_i$, $i = 0(1)N$ reduces (4.1.3) to a system of linear equations

$$(4.1.4)\quad Y_i - h \sum_{j=0}^{N} w_j\, k(x_i,t_j)\, Y_j = f(x_i), \qquad i = 0(1)N.$$

We denote the solution of (4.1.4) by $\bar{Y}^{(o)}$.

We now decide to include in (4.1.2) differences of order upto p $(p \leq N)$ as Gregory correction to the trapezoid rule, and let

$$(4.1.5)\qquad C_i^{(p)}(\bar{Y}) = h \sum_{s=1}^{p} c_s[\, \nabla^s \Theta_i(Y;b) + (-1)^s\, \Delta^s \Theta_i(Y;a)],$$

where $\Theta_i(Y;t) = k(x_i,t)Y(t)$. Then we solve the system

$$(4.1.6)\qquad Y_i^{(1)} - h \sum_{j=0}^{N} w_j\, k(x_i,t_j) Y_j^{(1)} = f(x_i) + C_i^{(p)}(\bar{Y}^{(o)}),$$

and get under certain conditions an improved solution $Y_i^{(1)}$. This process of obtaining $Y_i^{(1)}$ from $Y_i^{(o)}$ is called a deferred correction (see Baker[1977]).

The procedure above may be repeated, and for $r = 1, 2, \ldots,$ improved solution vectors $\bar{Y}^{(r)}$ be found iteratively from

$$(4.1.7) \qquad (I - K)\bar{Y}^{(r)} = \bar{f} + \bar{C}^{(p)}(\bar{Y}^{(r-1)}),$$

where $K = [hw_j \, k(x_i, t_j)]$, and

$$\bar{C}^{(p)}(\bar{Y}^{(r-1)}) = (C_0^{(p)}(\bar{Y}^{(r-1)}), \ldots, C_N^{(p)}(\bar{Y}^{(r-1)}))^T.$$

Since the sequence of the iterates $\bar{Y}^{(r)}$ is not guaranteed to converge, we take $Y^{(p)}$ as the final approximation. This technique is now commonly called Iterated Deferred Correction (IDC) (Baker[1977]).

A variant of this method is an IDC in which equation (4.1.7) is replaced by

$$(4.1.7^*) \qquad (I - K)\bar{Y}^{(r)} = \bar{f} + \bar{C}^{(r)}(\bar{Y}^{(r-1)}).$$

During the rth iteration of this method, the Gregory correction contains only differences of order upto r (a variable), and not upto p (a fixed constant) as in (4.1.7).

4.2 Asymptotic Expansion for IDC

We shall now derive an asymptotic expansion for the global error in the solution produced by the IDC method.

First, we derive, as in Section 3.2, an asymptotic expansion for the global error $\bar{y} - \bar{Y}^{(o)}$, where $\bar{Y}^{(o)}$ is the solution of

(4.2.1) $$(I - K)\bar{Y}^{(o)} = \bar{f}$$

and $\bar{y}$ is the solution of

(4.2.2) $$(I - K)\bar{y} = \bar{f} + \bar{\tau}$$

in which $\bar{\tau}$ is given by

(4.2.3) $$\tau_i \equiv E_i(y)$$
$$= \int_a^b k(x_i,t)\ y(t)\ dt - h \sum_{j=0}^{N} w_j k(x_i,t_j)\ y(t_j).$$

If we choose the w_j corresponding to the trapezoid rule, the quadrature errors τ_i have the form

(4.2.4) $$\tau_i = A_2(x_i)h^2 + A_4(x_i)h^4 + A_6(x_i)h^6 + \ldots .$$

Now subtraction of (4.2.1) from (4.2.2) yields

(4.2.5) $$(I - K)(\bar{y} - Y^{(o)}) = \bar{\tau}.$$

Consequently, the expansion for the global error is given by

(4.2.6) $$Y_i^{(o)} - y(x_i) = B_2^{(o)}(x_i)h^2 + B_4^{(o)}(x_i)h^4 + \ldots$$

Second, we derive an error expansion for the solution $Y_i^{(1)}$ obtained from $Y_i^{(o)}$ after a single iteration of deferred correction given by

(4.2.7) $$(I - K)\bar{Y}^{(1)} = \bar{f} + \bar{C}^{(p)}(\bar{Y}^{(o)}),$$

where $\bar{C}^{(p)}(\bar{Y}^{(o)})$ is the Gregory correction to the trapezoid rule consisting of differences of order upto p. A rearrangement of the correction terms in (4.1.5) yields the Lagrangian form:

(4.2.8) $$\bar{C}^{(p)}(\bar{Y}) = h \sum_{j=0}^{p} \omega_j^{(p)}[k(x_i, a+jh)\ Y(a+jh) + k(x_i, b-jh)\ Y(b-jh)].$$

The values of $\omega_j^{(p)}$ for p = 1(1)4 are tabulated below.

Table 4.2.1 Values of $\omega_j^{(p)}$

p \ j	0	1	2	3	4
1	-1/12	1/12			
2	-1/8	1/6	-1/24		
3	-109/720	177/720	-87/720	19/720	
4	-49/288	77/240	-7/30	73/720	-3/160

Using (4.2.6) and (4.2.8), we can write

(4.2.9) $$C_i^{(p)}(\bar{Y}^{(o)}) - C_i^{(p)}(\bar{y}) = A_2^{(o)}(x_i)h^3 + A_4^{(o)}(x_i)h^5 + \dots .$$

where $A_n^{(o)}(x_i) = \sum_{j=0}^{p} \omega_j^{(p)}[k(x_i, a+jh)\, B_n^{(o)}(a+jh)$

$+ k(x_i, b-jh)\, B_n^{(o)}(b-jh)], \qquad n = 2, 4, \ldots .$

If we now use the Euler-Maclaurin series formula and write τ_i as

$$(4.2.4) \qquad \tau_i = \sum_{j=1}^{\infty} \frac{B_{2j}}{(2j)!} h^{2j}(u_0^{(2j-1)} - u_N^{(2j-1)})$$

where the B's are Bernoulli constants (Grant[1969]), then the relation between the quadrature error and the Gregory correction may be expressed as in the following lemma.

Lemma 4.2.1

The Gregory correction $C_i^{(p)}(\bar{y})$ satisfies the equation

$$(4.2.10a) \qquad C_i^{(p)}(\bar{y}) = \tau_i - R_{p_0}(x_i) + O(h^{p+2}),$$

where

$$(4.2.10b) \quad R_{p_0}(x_i) = \sum_{j=0}^{\infty} \frac{B_{p_0+2j} h^{p_0+2j}}{(p_0+2j)!}[u_0^{(p_0+2j-1)} - u_N^{(p_0+2j-1)}],$$

with $u(t) = k(x_i, t)\, y(t)$ and $p_0 = p+2$ or $p+3$ according as p is even or odd. (This follows from the use of equations (4.1.[illegible]) and (4.2.4), and formulae (page 118, Baker 1977) for the derivatives in terms of differences of order p at most.)

We observe that the $O(h^{p+2})$ in (4.2.10a) is the error

In the above lemma, if τ_1 is given by (4.2.4), the series $R_{p_o}(x_1)$ may be written as

$$(4.2.10c) \quad R_{p_o}(x_1) = A_{p_o}(x_1)h^{p_o} + A_{p_o+2}(x_1)h^{p_o+2} + \dots .$$

But we can write

$$(4.2.11) \quad C_1^{(p)}(\bar{Y}^{(o)}) = C_1^{(p)}(\bar{Y}^{(o)}) - C_1^{(p)}(\bar{y}) + C_1^{(p)}(\bar{y})$$

$$= (A_2^{(o)}(x_1)h^3 + A_4^{(o)}(x_1)h^5 + \dots)$$

$$+ \tau_1 - R_{p_o}(x_1) + O(h^{p+2}),$$

from (4.2.9) and (4.2.10a). This gives

$$(4.2.11a) \qquad C_1^{(p)}(\bar{Y}^{(o)}) = \tau_1 + \sigma_1^{(1)},$$

where

$$(4.2.11b) \qquad \sigma_1^{(1)} = (A_2^{(o)}(x_1)h^3 + A_4^{(o)}(x_1)h^5 + \dots)$$

$$- R_{p_o}(x_1) + O(h^{p+2}).$$

Therefore (4.2.7) can be written as

$$(4.2.12) \qquad (I - K)\bar{Y}^{(1)} = \bar{f} + \bar{\tau} + \bar{\sigma}^{(1)}$$

Subtracting (4.2.2) from the above equation, we have

$$(I - K)(\bar{Y}^{(1)} - \bar{y}) = \bar{\sigma}^{(1)},$$

and the error expansion can be written in the form

$$(4.2.13)\quad Y_1^{(1)} - y(x_1) = (B_2^{(1)}(x_1)h^3 + B_4^{(1)}(x_1)h^5 + \ldots) + S_{p_0}^{(1)}(x_1) + O(h^{p+2}),$$

where $S_{p_0}^{(1)}$ stands for the same series as R_{p_0} but with different coefficients.

Third, we give the asymptotic behaviour of an iterated deferred correction procedure in the following theorem.

Theorem 4.1

Let $\bar{Y}^{(o)}$ be the approximation to the solution of FIE (4.1.1) by the trapezoid rule with stepsize h, and let the vectors $\bar{Y}^{(r)}$, $r = 1(1)p$ satisfying

$$(4.2.14)\qquad (I - K)\bar{Y}^{(r)} = \bar{I} + \bar{C}^{(p)}(\bar{Y}^{(r-1)})$$

represent the successive approximations by the method of IDC using differences of order upto p in the Gregory Correction to the trapezoid rule. Let $p_0 = p+2$ or $p+3$ according as p is even or odd, and suppose the p_0th t-derivatives of the functions $k(x_1,t)\ y(t)$ are continuous in $[a,b]$. Then the asymptotic expansion for the global error is given by

$$(4.2.15)\quad Y_1^{(r)} - y(x_1) = (B_2^{(r)}(x_1)h^{r+2} + B_4^{(r)}(x_1)h^{r+4} + \ldots) + S_{p_0}^{(r)}(x_1) + O(h^{p+2}),$$

where $S_{p_0}^{(r)}(x_1)$ is of the form R_{p_0} of (4.2.10).

Proof

We have already shown that the theorem is true for $r = 1$. Suppose it is true for $r = 1, 2, \dots, r_0 (r_0 < p)$. Then (4.2.15) is true, in particular for $r = r_0$. That is,

$$(4.2.16) \quad Y_1^{(r_0)} - y(x_1) = (B_2^{(r_0)}(x_1)h^{r_0+2} + B_4^{(r_0)}(x_1)h^{r_0+4} + \dots) + S_{p_0}^{(r_0)}(x_1) + O(h^{p+2}).$$

We shall now prove that (4.2.15) is true for $r = r_0+1$.

The equation (4.2.16) implies

$$\sigma_1^{(p)}(\bar{Y}^{(r_0)}) - \sigma_1^{(p)}(\bar{y}) = (A_2^{(r_0)}(x_1)h^{r_0+3} + A_4^{(r_0)}(x_1)h^{r_0+5} + \dots) + T_{p_0}^{(r_0)}(x_1) + O(h^{p+2}), \text{ where}$$

$$A_n^{(r_0)}(x_1) = \sum_{j=0}^{p}{}' \Omega_j^{(p)} [k(x_1, a+jh)\, B_n^{(r_0)}(a+jh) + k(x_1, b-jh)\, B_n^{(r_0)}(b-jh)],$$

and $T_{p_0}^{(r_0)}(x_1)$ has the same form as R_{p_0}.

Again using (4.2.10) we can write

$$(I - K)(\bar{Y}^{(r_0+1)} - \bar{y}) = \bar{\sigma}^{(r_0+1)}, \text{ where}$$

$$\sigma_1^{(r_0+1)} = (A_2^{(r_0)}(x_1)h^{r_0+3} + A_4^{(r_0)}(x_1)h^{r_0+5} + \dots) + T_{p_0}^{(r_0)}(x_1) - R_{p_0}(x_1) + O(h^{p+2}).$$

Now we can let

$$S_{p_o}^{(r_o+1)}(x_1) = T_{p_o}^{(r_o)}(x_1) - R_{p_o}(x_1),$$

since each series is of the same form as $R_{p_o}(x_1)$.

Therefore, we have

$$Y_1^{(r_o+1)} - y(x_1) = (B_2^{(r_o+1)}(x_1)h^{r_o+3} + B_4^{(r_o+1)}(x_1)h^{r_o+5} + \dots) + S_{p_o}^{(r_o+1)}(x_1) + O(h^{p+2}).$$

Hence the theorem is true when $r = r_o+1$, i.e. for $r = 1(1)\, r_o + 1 \leq p$.

4.3 Iterated Deferred Defect Correction (IDDeC)

4.3.1 Description of the method of IDDeC

We now describe an application of the method of IDeC, which was explained in Section 2.2, on the method of IDC.

We assume that N is a multiple of m, where m is even and greater than max[p, 2], with the meaning of p as in (4.1.5). Let $Z_1^{(o,p)}$ denote the IDC solution Y_1 of the FIE (4.1.1).

We divide [a,b] into subintervals $I_j = [x_{(j+1)m}, x_{jm}]$, $j = 1(1)N/m$. For each j, let $P_{j,m}^{(o)}(x)$ be the polynomial of degree m interpolating the m+1 points $(x_1, Z_1^{(o,p)})$,

$(j-1)m \le i \le jm$. We then define, for x in $[a,b]$, the interpolating function $P_m^{(o)}(x)$ as the collection of polynomials $P_{j,m}^{(o)}$ such that

$$(4.3.1) \qquad P_m^{(o)}(x) = P_{j,m}^{(o)}(x) \text{ for } x \in I_j, \quad j = 1(1)N/m.$$

Consider now the problem

$$(4.3.2) \quad y(x) - \int_a^b k(x,t)\, y(t)\, dt = g^{(o)}(x), \qquad a \le x \le b$$

where

$$g^{(o)}(x) = P_m^{(o)}(x) - \int_a^b k(x,t)\, P_m^{(o)}(x)\, dt.$$

Since $P_m^{(o)}(x)$ is a good approximation to $y(x)$, the expression $g^{(o)}(x) - f(x)$ is expected to be small and thus (4.3.2) is a neighbouring problem (NP) of our original problem (OP) (4.1.1). The exact solution of this new problem is the known function $P_m^{(o)}(x)$. We solve this NP using the same method by which we solved our OP, and denote the resulting solution by $\gamma_i^{(o,p)}$. Its known error $P_m^{(o)}(x_i) - \gamma_i^{(o,p)}$ is taken as an estimate for the unknown error $y(x_i) - Z_i^{(o,p)}$ in the IDC solution $Z_i^{(o,p)}$ of our OP. Thus, to the true solution $y(x)$, the approximation $Z_i^{(1,p)}$ given by

$$(4.3.4) \qquad Z_i^{(1,p)} = Z_i^{(o,p)} + P_m^{(o)}(x_i) - \gamma_i^{(o,p)}$$

is more accurate than $Z_i^{(o,p)}$.

We repeat the process above, and find improved approximations $Z_i^{(s+1)}$, $s = 0, 1, 2, \ldots$, given by

$$(4.3.5) \qquad Z_i^{(s+1,p)} = Z_i^{(o,p)} + P_m^{(s)}(x_i) - \gamma_i^{(s,p)},$$

where the function $P_m^{(s)}(x)$, defined analogous to the function $P_m^{(o)}(x)$ of (4.3.2), interpolates the points $(x_j, Z_i^{(s,p)})$, and $\gamma_i^{(s,p)}$ is the IDC solution of the following NP:

$$(4.3.6) \quad y(x) - \int_a^b k(x,t)\, y(t)\, dt = g^{(s)}(x), \qquad a \leq x \leq b,$$

with $$g^{(s)}(x) = P_m^{(s)}(x) - \int_a^b k(x,t)\, P_m^{(s)}(t)\, dt.$$

Thus, starting with the solution $Z_i^{(o,p)}$ produced by the IDC, we find its iterative improvements by the method of IDeC, each iteration of which involves an IDC. Hence we call the entire technique the method of Iterated Deferred Defect Correction (IDDeC).

4.3.2 Asymptotic expansion for IDDeC

We recall the steps carried out during the first iteration of our IDDeC. First, the solutions of our OP and NP are obtained as depicted in Table 4.3.1. Second, the improved approximation $Z_i^{(1,p)}$ of $Z_i^{(o,p)} \equiv Y_i^{(p)}$ is computed using (4.3.4).

We now proceed to derive an asympt

the global error in the first iterate Z

Theorem 4.1, the solution $Z_1^{(o,p)}$ found

satisfies the equation

$$(4.3.7) \quad Z_1^{(o,p)} - y(x_1) = (B_2^{(p)}(x_1)h^{p+2}$$

$$+ S_{p_o}^{(p)}(x_1) +$$

where $S_{p_o}^{(p)}(x_1) = \alpha_{p_o}^{(p)}(x_1)h^{p_o} + \alpha_{p_o+2}^{(p)}h$

is either the integers p+2 or p+3 wh

Table 4.3.1 Key steps

Problem	Method	System solved
OP	trapezoid	$(I-K)\bar{Y}^{(o)} = \bar{f}$
	IDC with r=1(1)p	$(I-K)\bar{Y}^{(r)} = \bar{f} + \mho^{(p)}(\bar{Y}$
NP	trapezoid	$(I-K)\bar{\gamma}^{(o,o)} = \bar{g}^{(o)}$
	IDC with r=1(1)p	$(I-K)\bar{\gamma}^{(o,r)} = \bar{g}^{(o)} + \mho^{(p)}(\bar{\gamma}$

We now assume p is even. Since $P_m^{(o)}(x)$ interpolates the points $(x_1, Z_1^{(o,p)})$, we have

$$(4.3.8) \qquad P_n^{(o)}(x) = y(x) + B_2^{(p)}(x)h^{p+2} + \dots$$
$$+ S_{p+2}^{(p)}(x_1) + O(h^{p+2}),$$

provided $p+2 \leq m+2$, and the error due to interpolation is assumed to be included in the $O(h^{p+2})$ term. Therefore the error

$$\tau_1^{(o)} \doteq E_1(P_m^{(o)})$$

in the solution of NP (4.3.1) can be written as

$$(4.3.9) \qquad \tau_1^{(o)} = \tau_1 + O(h^{p+4}).$$

Analogous to the equations (4.2.5) and (4.2.6) in the case of OP, the equations for our NP are

$$(4.3.10) \qquad (I - K)(\bar{P}_m^{(o)} - \bar{\gamma}^{(o,o)}) = \bar{\tau}^{(o)},$$

and

$$(4.3.11) \quad \gamma_1^{(o,o)} - P_m^{(o)}(x_1) = B_2^{(o)}(x_1)h^2 + \dots$$
$$+ B_{p+2}^{(o)}(x_1)h^{p+2} - D_{p+4}^{(o)}(x_1)h^{p+4} + \dots .$$

Therefore,

$$(4.3.12)\quad C_i^{(p)}(\bar{\gamma}^{(o,o)}) - C_i^{(p)}(\bar{P}_m^{(o)}) = A_2^{(o)}(x_i)h^3 + \ldots$$

$$+ A_{p+2}^{(o)}(x_i)h^{p+3} + D'^{(o)}(x_i)h^{p+4} + \ldots .$$

But we have

$$(4.3.13)\qquad C_i^{(p)}(\bar{P}_m^{(o)}) = \tau_i^{(o)} - R_{p+2}(x_i) + O(h^{p+2})$$

and (4.3.12) becomes

$$(4.3.14)\quad C_i^{(p)}(\bar{v}^{(o,o)}) = (A_2^{(o)}(x_i)h^3 + \ldots + A_{p+2}^{(o)}(x_i)h^{p+3}$$

$$+ D'^{(o)}_{p+4}(x_i)h^{p+5} + \ldots) + \tau_i^{(o)} - R_{p+2}(x_i) + O(h^{p+2})$$

An application of IDC to NP gives rise to

$$(4.3.15)\qquad (I-K)\bar{\gamma}^{(o,1)} = \bar{g}^{(o)} + \bar{\tau}^{(o)} + \bar{\sigma}^{(o,1)},$$

where

$$\sigma_1^{(o,1)} = (A_2^{(o)}(x_1)h^3 + \ldots + A_{p+2}^{(o)}(x_i)h^{p+3} +$$

$$+ D'^{(o)}_{p+4}(x_i)h^{p+5} + \ldots) + R_{p+2}(x_i) + O(h^{p+2}).$$

Moreover,

$$(I-K)(\bar{\gamma}^{(o,1)} - \bar{P}_m^{(o)}) = \bar{\sigma}^{(o,1)}$$

and

$$(4.3.16)\quad \gamma_1^{(o,1)} - P_m^{(o)}(x_i) = (B_2^{(1)}(x_i)h^3 + \ldots$$

$$+ B_{p+2}^{(1)}(x_1)h^{p+3} + D_{p+4}^{(1)}(x_i)h^{p+4} + \ldots) + S_{p+2}^{(1)}(x_i)$$

After r applications of deferred correction, we have

$$(4.3.17)\quad Y_1^{(o,r)} - P_m^{(o)}(x_1) = (B_2^{(r)}(x_1)h^{2+r} + \dots$$

$$+ B_{p+2}^{(r)}(x_1)h^{p+2+r} + D_{p+4}^{(r)}(x_1)h^{p+4+r} + \dots)$$

$$+ S_{p+2}^{(r)}(x_i) + O(h^{p+2}),$$

provided $p+4+r \leq m+2$.

We note that the term $O(h^{p+2})$ in (4.3.17) is the same as that in (4.2.15), except for the error due to interpolation. Using Zadunaisky's principle, we therefore have the result:

if $2p+4 \leq m+2$, then

$$(4.3.18)\qquad Z_1^{(1,p)} = y(x_1) + B_{p+4}^{(1,p)}(x_1)h^{2p+4}$$

$$+ B_{p+6}^{(1,p)}(x_1)h^{2p+6} + \dots$$

A similar result holds when p is odd, and the following theorem is true, by induction.

Theorem 4.2

Under the conditions stated in Theorem 4.1, for each $s = 0, 1, 2, \dots$ the solution $Z_i^{(s,p)}$ produced at the end of the sth iteration of the IDDeC, the NPs of which involve polynomials of even degree m, has an expansion of the form:

$$(4.3.19) \qquad Z_1^{(s,p)} = y(x_1) + B_q^{(s,p)}(x_1)h^q + B_{q+2}^{(s,p)}(x_1)h^{q+2} + \dots$$

where

$$(4.3.19a) \qquad q = \min[(s+1)(p+2),\ m+2].$$

Remarks

From Theorem 4.2 above, it is seen that when $s=0$ our IDDeC reduces to an IDC, and the maximum order of accuracy obtainable is $p+2$ (cf. Theorem 4.22 in Baker[1977]). On the other hand when $p=0$. our IDDeC degenerates into an IDeC, and the maximum order of accuracy q_{max} attainable after s iterations is $\min[2(s+1),\ m+2]$. In general, if the order of the underlying method is n, then $q_{max} = \min[(s+1)n,\ m+2]$. Thus, our results agree with those concerning the IDeC methods applied to differential and integral equations, given respectively by Frank[1975, 1976, 1977] and Sathiaraj and Sankar[1983] (see Theorem 3.2.1), and IDDeC is indeed a generalization of both IDC and IDeC.

4.3.3 Implementation of IDDeC

We give below an outline of the IDDeC as applied to (4.1.1).

Algorithm IDDeC

1. Input a, b, N, h, p, and m.

2. Find the solution of the OP by the use of a basic

```
    discretization rule and then find the solution $Z_i$
    by IDC using upto the pth order differences as
    correction terms to the rule.  Copy $Z_i$ in $Z_i^{OP}$.
3. For $s = 1, 2, \ldots,$ do
    3.1 Find a piecewise interpolating polynomial
        $P_m(x)$ of degree m fitting the data points
        $(x_i, Z_i)$.
    3.2 Solve the NP first by the basic rule and then
        by the same IDC method used in Step 2.  Let
        its solution be $Z_i^{NP}$.
    3.3 Improve the solution, using
        $Z_i := Z_i^{OP} + (Z_i - Z_i^{NP})$
4. Output $Z_i$, $i = 0(1)N$.
```

In Step 3.2, the solution of NP involves the calculation for $i = 0(1)N$, of the values

$$I_i = \int_a^b k(x_i,t)P_m(t)dt = \sum_{j=1}^{N/m} \int_a^b k(x_i,t)P_{m,j}(t)dt. \tag{4.3.20}$$

For each j, we find the polynomial $P_{m,j}$ and then store the corresponding values of

$$\int_a^b k(x_i,t)P_{m,j}(t)dt$$

in an array called VALINT (i,j), $i = 0(1)N$, $j = 1(1)N/m$. Then the values I_i are found using

$$I_i = \sum_{j=1}^{N/m} \text{VALINT } (i,j).$$

We made use of the NAG routines while calculating the integrals in (4.3.20).

4.4 Applications to Integral Equations

This section deals with applications of the methods of iterated deferred correction and iterated deferred defect correction to illustrative examples in linear Fredholm, and linear and nonlinear Volterra integral equations.

4.4.1 Linear Fredholm Equations

We implemented the method of IDDeC as applied to (4.1.1). The techniques of IDDeC and IDC were based on quadrature methods which make use of the trapezoid and Simpson's rule.

(a) The IDC methods based on the trapezoid rule

We give results from the computational tests performed using 16-18 significant digits. Problems 1 to 4 of Section 2.2.2 were solved by IDCs based on the trapezoid rule together with the Gregory correction and by IDDeCs generated by the above IDCs.

We denote by IDC(p,s) the IDC method involving s iterations of deferred correction with upto pth order correction terms to the trapezoid rule. Similarly the notation IDDeC(p,s) stands for the IDDeC method which involves s iterations of defect correction and which is based on the method IDC(p,p).

The problems were solved by these methods for different values of p, s and N; the degree m of the polynomials used in IDDeC was chosen to be 4, 6, ..., and 14. In the following tables we display the maximum absolute errors in the computed solutions and the errors with asterisks (*) indicate that they remain the same for the subsequent iterations/corrections.

We tested the performance of the two types of IDC defined by (4.1.7) and (4.1.7') and found them essentially the same (see Theorem 4.22 in Baker[1977]). Table 4.4.1 displays the results for Problem 1 by these two types of IDC with Gregory correction $C^{(q)}$, where q is a constant p or a variable r, and those by the corresponding IDDeC methods. We wish to add that the methods IDDeC(4,1) and IDDeC(3,1) yielded the same results as IDDeC(2,2), whether q is a constant or a variable.

We also tested an IDC method in which the first deferred correction involved differences of first order only, but the second correction involved differences of upto the third order. The IDDeC method based on the above IDC was also implemented. The results of these methods are listed in Table 4.4.2.

Table 4.4.1 Errors in solution to Problem 1 by IDC and IDDeC with correction $\bar{C}^{(q)}$ where q is a constant p = 4 or a variable r.

q	N	IDC Iteration number r					IDDeC	
		0	1	2	3	4	(2,1)	(2,2)
Constant	8	8.7E-3	1.2E-3	6.0E-5	1.4E-5	5.1E-7	1.2E-11	1.5E-11*
Variable		8.7E-3	1.2E-3	5.6E-5	1.4E-5	4.6E-7	1.2E-10	1.5E-11*
Constant	16	2.2E-3	1.5E-4	4.2E-6	4.5E-7	1.1E-8	1.2E-14	1.5E-14*
Variable		2.2E-3	1.5E-4	4.1E-6	4.5E-7	1.1E-8	2.6E-13	1.5E-14*
Constant	32	5.4E-4	1.9E-5	2.8E-7	1.4E-8	2.0E-10	1.0E-17	1.5E-17*
Variable		5.4E-4	1.9E-5	2.8E-7	1.4E-8	2.0E-10	5.2E-16	1.5E-17

Table 4.4.2 IDC, and IDDeC with different values of m

		IDC			IDDeC	
	N	(1,0)	(1,1)	(3,2)	m	Iteration 1
	12	3.9E-3	3.5E-4	1.3E-6	4	1.2E-8*
					6	5.5E-11*
Problem 1	24	9.6E-4	4.5E-5	4.1E-8	4	1.9E-10*
					6	2.1E-13*
					8	2.7E-16*
	48	2.4E-4	5.7E-6	1.3E-9	4	2.9E-12*
					6	8.4E-16*
					8	4.3E-18*
	12	2.7E-3	3.0E-5	3.5E-6	4	3.4E-7*
					6	1.2E-8*
Problem 2	24	6.8E-4	1.9E-6	5.6E-8	4	5.0E-9*
					6	4.5E-11*
					8	4.6E-13*
	48	1.7E-4	1.2E-7	8.8E-10	4	8.0E-11*
					6	1.7E-13*
					8	4.4E-16*

Table 4.4.3 IDC with q = r = 1 & 2, and IDDeC with m = 4

Problem	N	IDC(1,1)	IDDeC(1,1)	IDDeC(1,2)	IDC(2,2)	IDDeC(2,1)
	12	3.5E-4	3.8E-8	1.2E-8*	1.2E-5	1.2E-8*
1	24	4.5E-5	6.2E-10	1.9E-10*	1.2E-7	1.9E-10*
	48	5.7E-6	9.8E-12	2.9E-12*	5.6E-8	2.9E-12*
	12	3.0E-5	3.3E-7	3.4E-7*	5.8E-5	3.4E-7*
2	24	1.9E-6	5.1E-9*		3.7E-6	5.1E-9*
	48	1.2E-7	7.9E-11	8.0E-11*	2.3E-7	8.0E-11*

Next, we list in Table 4.4.3 the results by the IDC and IDDeC methods for which the maximum order of differences q in the Gregory correction term $\bar{C}^{(q)}$ is taken to be the i~~n~~tera~~c~~tion number, r(r = 1, 2). We find that even for the small value 4 of m the IDDeC produces very good results.

The numerical results for the two types of IDC methods, with q = p or r, are essentially the same. This is true also for the corresponding IDDeC methods which are generated by those IDCs. Therefore we list in the succeeding Tables 4.4.4 to 4.4.8 the results only of the first version of the IDC method, and those of the corresponding IDDeC method.

Table 4.4.4a IDC and IDDeC methods with p = 1

				IDDeC iteration number		
Problem	m	N	IDC	1	2	3
1	6	6	2.7E-3	3.1E-6	1.7E-8	1.4E-8
		12	3.5E-4	5.0E-8	6.2E-11	5.5E-11
		24	4.5E-5	8.1E-10	2.3E-11	2.1E-13
	8	10	1.2E-3	5.6E-7	2.8E-10	1.5E-11
		16	1.5E-4	9.1E-9	5.6E-13	1.5E-14
		32	1.9E-5	1.4E-10	1.1E-15	1.5E-17
	10	10	6.1E-4	1.5E-7	3.6E-11	3.9E-15
		20	7.7E-5	2.4E-9	7.4E-14	5.3E-18
		40	9.8E-6	3.8E-11	1.5E-16	4.3E-18
2	6	6	4.8E-4	4.8E-6	4.3E-6	4.5E-6
		12	3.0E-5	1.7E-8	1.2E-8*	
		24	1.9E-6	7.8E-11	4.5E-11*	
	8	8	1.5E-4	3.3E-8	3.9E-8	NC
		16	9.4E-5	5.4E-10	2.8E-11	
		32	5.9E-7	4.4E-12	2.6E-14	
3	6	6	2.4E-3	5.7E-5	1.2E-6	2.3E-8
		12	3.3E-4	1.2E-6	3.5E-9	1.1E-11
		24	4.4E-5	2.2E-8	8.9E-12	3.7E-15
	8	8	1.1E-3	1.2E-5	1.1E-7	NC
		16	1.4E-4	2.3E-7	3.0E-10	
		32	1.9E-5	4.0E-9	7.2E-13	
4	6	6	3.0E-3	8.8E-5	1.6E-5	1.8E-5
		12	3.7E-4	1.4E-6	7.3E-8	7.7E-8
		24	4.7E-5	2.5E-8	2.9E-10	3.0E-10
	8	8	1.2E-3	1.4E-5	2.9E-7	NC
		16	1.6E-4	2.6E-7	5.4E-10	
		32	2.0E-5	4.6E-9	1.0E-12	

Note: NC stands for 'not computed'.

Table 4.4.4b Computed orders for the iterates of IDDeC with p = 1

		IDDeC iteration			
Problem	m	0	1	2	3
1	6	3	6	8	8
	8	3	6	9	10
	10	3	6	9	L
2	6	4	6+	8	8
	8	4	6+	10	NC
3	6	3	6	8+	10+
	8	3	6	9	NC
4	6	3	6	8	8
	8	3	6	9	NC

Table 4.4.4b Computed orders for the iterates of IDDeC with p = 1

		IDDeC iteration			
Problem	m	0	1	2	3
1	6	3	6	8	8
	8	3	6	9	10
	10	3	6	9	L
2	6	4	6+	8	8
	8	4	6+	10	NC
3	6	3	6	8+	10+
	8	3	6	9	NC
4	6	3	6	8	8
	8	3	6	9	NC

Tables 4.4.4 to 4.4.6 give the maximum errors of the methods IDC(p,p) and IDDeC(p,s) for various values of p, s and N when applied to Problems 1-4.

Table 4.4.4a gives the errors in the solution for Problems 1 to 4 by the IDC method with p = 1 and by the IDDeC method with different values of m, and Table 4.4.4b depicts the computed orders of convergence for the iterates of the IDDeC methods. These computed orders are in accordance with the theoretical orders predicted by the equation (4.3.19a). In the table a number with a '+' sign means the computed order is more than that number.

Table 4.4.5 portrays the validity of order relation (4.3.19a), for m = 6(2)10 and p = 2(1)4 in the case of Problem 1, and Table 4.4.6 for m = 6 and 8, and p = 2 in the case of Problems 2 to 4.

Tables 4.4.7 and 4.4.8 display the results by the IDC and IDDeC methods for different values of p, and reveals the power of IDDeC over a wide range of m and also for relatively small values of N.

Our computational tests show that the IDDeC method is superior to the IDeC as regards accuracy and cost.

Table 4.4.5 The errors for Problem 1 by IDC(p,p) and IDDeC(p,1) where p = 2, 3, 4.

	N	m = 6		m = 8		m = 10	
		IDC	IDLeC	IDC	IDDeC	IDC	IDDeC
p=2	m	2E-4	1E-8*	6E-5	1E-11*	3E-5	4E-13*
	2m	1E-5	5E-11*	4E-6	1E-14*	2E-6	4E-16*
	4m	9E-7	2E-13*	3E-7	1E-17*	1E-7	4E-18*
Order		4	8	4	10	4	8
p=3	m	6E-5	1E-8*	1E-5	2E-11*	5E-6	1E-14*
	2m	2E-6	6E-11*	4E-7	2E-14*	2E-7	4E-18*
	3m	6E-8	2E-13*	1E-8	2E-17*	5E-9	4E-18*
Order		5	8	5	10	5	10+
p=4	m	2E-6	1E-8*	5E-7	2E-11*	2E-7	1E-14*
	2m	6E-8	6E-11*	1E-8	2E-14*	3E-9	3E-18*
	3m	1E-9	3E-13*	2E-10	2E-17*	5E-11	4E-18*
Order		6	8	6	10	6	12

Table 4.4.6 IDC(2,2) and IDDeC with m = 6 & 8

		m = 6			m = 8		
	N	IDC	IDDeC iteration 1	2	IDC	IDDeC iteration 1	2
Problem 2	m	8.5E-4	4.2E-6*		2.8E-4	3.9E-8*	
	2m	5.8E-5	1.2E-8*		1.8E-5	2.8E-11*	
	4m	3.7E-6	4.5E-11*		1.2E-6	2.6E-14*	
Order		4	8*		4	10*	
Problem 3	m	2.4E-4	3.2E-6	2.8E-8	7.1E-5	4.0E-7	1.3E-9
	2m	1.4E-5	1.9E-8	1.4E-11	4.4E-6	2.2E-9	5.3E-13
	4m	8.8E-7	9.6E-11	5.0E-15	2.8E-7	1.0E-11	1.8E-16
Order		4	8	8+	4	8	10+
Problem 4	m	1.9E-3	1.1E-5	1.8E-5	7.0E-4	1.1E-6	1.6E-7
	2m	1.5E-4	2.8E-8	7.7E-8	5.1E-5	5.8E-9	1.9E-10
	4m	1.1E-5	4.5E-11	3.0E-10	3.4E-6	2.7E-11	1.9E-13
Order		4	8+	8	4	8	10

Table 4.4.7 IDC and IDDeC methods with p = 2 and 4

				m = 12		m = 14	
	p	N	IDC	IDDeC		IDDeC	
			(p,p)	(p,1)	(p,2)	(p,1)	(p,2)
Problem 2	2	m	5.8E-5	1.8E-12	1.5E-12*	5.9E-14	6.3E-1
		2m	3.7E-6	3.7E-16	6.6E-17*	6.5E-17	3.9E-1
Order				8+	14+	8+	L
	4	m	2.0E-6	1.5E-12*		6.3E-15*	
		2m	3.3E-8	6.8E-17*		1.5E-18	3.7E-1
Order				12+	14+	12+	L
Problem 3	2	m	1.4E-5	1.9E-8	1.4E-11	6.0E-9	2.4E-1
		2m	8.8E-7	9.6E-11	5.0E-15	2.9E-11	8,3E-1
Order				8	12	8	12
	4	m	5.3E-8	1.8E-11	7.2E-16	3.5E-12	7.2E-17
		2m	1.0E-9	8.7E-15	3.9E-18	1.5E-15	7.8E-1
Order				12	L	12	L
Problem 4	2	m	1.5E-4	4.9E-8	3.5E-11	1.5E-8	7,3E-12
		2m	1.1E-5	2.5E-10	1.5E-14	7.7E-11	2.6E-15
Order				8	12	8	12
	4	m	4.9E-7	3.2E-10	6.2E-12	5.6E-11	2.8E-14
		2m	1.2E-8	1.1E-13	5.2E-16	1.9E-14	4.8E-18
Order				12	14	12	L

Table 4.4.8 Errors in solutions by IDDeC

		m = 4		m = 12	
		IDDeC		IDDeC	
Problem	N	(2,1)	(2,2)	(3,1)	(3,2)
1	m	8.4E-6*		2.3E-17	8.7E-18
	2m	1.4E-7*		2.2E-18	4.3E-18
2	m	3.4E-4*		1.5E-12	1.5E-12
	2m	4.0E-6*		6.9E-17	6.7E-17
3	m	5.6E-5	1.8E-6	8.6E-10	1.5E-13
	2m	3.8E-7	1.2E-9	1.2E-12	1.1E-17
4	m	1.5E-3	1.4E-3	3.5E-9	5.5E-12
	2m	2.4E-5	2.3E-5	5.2E-12	4.8E-16

(b) The IDC methods based on Simpson's formula

Just as Gregory's formula represents the trapezoid rule plus a correction, the following equation represents

Simpson's rule with a correction (see Fox[1962]):

$$(4.4.1)\qquad \int_{x_0}^{x_N} u\,dx = \frac{h}{3}\,(u_0 + 4u_1 + 2u_2 + \dots + 4u_{N-1} + u_N)$$

$$- \frac{h}{90} \sum_{j=1(2)}^{N-1} \delta^4 u_j + \dots,$$

where N is even and

$$(4.4.2)\qquad \delta^4 u_j = u_{j+2} - 4u_{j+1} + 6u_j - 4u_{j-1} + u_{j-2}.$$

Let us apply the quadrature method based on Simpson's rule to the FIE (4.1.1), and obtain the solution $Y_1^{(o)}$ from

$$(4.4.3)\qquad (I-K)\bar{Y}^{(o)} = \bar{f}$$

where K is analogous to the matrix in (4.1.7).

With the notation of Section 4.1, a single application of deferred correction gives $Y_1^{(1)}$ from

$$(4.4.4a)\qquad (I-K)\bar{Y}^{(1)} = \bar{f} + \bar{C}^{(4)}(\bar{Y}^{(o)}),\ \text{where}$$

$$(4.4.4b)\qquad C_1^{(4)}(\bar{Y}) = -\frac{h}{90}\,[\delta^4 Q_1(Y;\ t_1) + \delta^4 Q_1(Y;\ t_3) + \dots$$

$$+ \delta^4 Q_1(Y;\ t_{N-1})]$$

with $Q_1(Y;\ t) = k(x,t)Y(t)$.

We now consider an IDC method defined by

$$(4.4.5)\qquad (I-K)\bar{Y}^{(r)} = \bar{f} + \bar{C}^{(4)}(\bar{Y}^{(r-1)}),\qquad r = 1,\ 2.$$

a knowledge of $k(x_i,t)Y(t)$ and $f(x)$ for points $t = x = x_{-1}$ and x_{N+1}, which are outside the basic interval. But then we can compute the values outside the range with the use of (4.1.1) or

$$(4.4.7) \qquad Y^{(o)}(x) - S(x) = f(x),$$

where $S(x)$ represents the Simpson formula:

$$S(x) = \tfrac{h}{3}[k(x,x_0)Y_0^{(o)} + 4k(x,x_1)Y_1^{(o)} + 2k(x,x_2)Y_2^{(o)} + \dots$$

$$+ 4k(x,x_{N-1})Y_{N-1}^{(o)} + k(x,x_N)Y_N^{(o)}]$$

Then the use of (4.4.4a) gives $Y_i^{(1)}$, $i = 0(1)N$. Again the corrected values $Y_{-1}^{(1)}$ and $Y_{N+1}^{(1)}$ may be found from

$$(4.4.8) \qquad Y^{(r)}(x_i) - S(x_i) = f(x_i) + C_i^{(4)}(Y^{(r-1)})$$

with $i = -1$ and $N+1$. Similarly the corrected values $Y_i^{(2)}$, $i = -1(1)N+1$ may then be found using (4.4.5) and (4.4.8).

Our IDDeC proceeds by taking $Z_i^{'(o,2)} \equiv Y_i^{(2)}$ and finding the solution of the sth NP as we have done in the case of OP. As for the evaluation of $g^{(r)}(x)$ at $x = x_{-1}$ and x_{N+1}, we assume $P_m^{(s)}(x_i) = Y_i^{(s)}$, $i = -1, N+1$.

Numerical results

Problems 1 to 4 of Section 2.2.2 were solved by the IDC and IDDeC methods based on Simpson's rule with correction (4.4.1). In Table 4.4.9 we list the errors in

Table 4.4.9 Results by IDC and IDDeC methods based on Simpson's rule with correction

Problem	N	m = 8			m = 10		m = 12	
			IDDeC		IDDeC		IDDeC	
		IDC(2)	1	2	1	2	1	2
1	m	4.21E-8	1E-10	2E-11	2E-11	3E-14	6E-12	1E-14
	2m	6.67E-10	8E-13	1E-14	2E-13	1E-14	4E-14	3E-17
2	m	1.11E-6	4E-8*		3E-10*		2E-10	1E-12
	2m	1.82E-8	5E-11	3E-11	4E-12	5E-14	1E-12	8E-16
3	m	3.80E-8	8E-10	1E-11	1E-10	1E-12	3E-11	2E-13
	2m	6.05E-10	3E-12	2E-14	6E-13	2E-15	2E-13	5E-16
4	m	6.79E-7	2E-7*		9E-10	1E-9	1E-10	5E-12
	2m	1.12E-8	2E-10*		5E-12	4E-13	2E-12	4E-15

In our implementation of IDDeC, we solved the NPs by taking $g^{(s)}(x_i) = f(x_i)$, $i = -1, N+1$. Moreover, we assumed

$$Y_i^{(2)} = y(x_i), \qquad i = -1, N+1.$$

Our computational results indicate that the order of the IDC(2) method is 6 as expected. The IDDeC method produces highly accurate results but the order of accuracy is not as high as predicted by the theoretical analysis.

We note that the IDDeC method based on the trapezoid rule with correction upto fourth order differences as well as the one based on Simpson's rule with a correction (4.4.1) is expected to produce solutions of the same order, namely 12, 18, ..., or m+2. However, we would rather choose the former because of the simplicity of the trapezoid rule and because of the implementation difficulties associated with the latter IDDeC. (Compare results for m = 12 in Tables 4.4.7 - 4.4.9.)

4.4.2 Linear Volterra Integral Equations

In Section 2.4 we considered the quadrature methods for linear Volterra integral equations of the second kind

$$(4.4.9) \qquad y(x) - \int_0^x k(x,t)\, y(t)\, dt = f(x), \qquad 0 \leq x \leq a.$$

Now we shall describe two approaches of iterated deferred correction (IDC) based on a quadrature method. We tested the performances of these IDCs and the generated IDDeCs.

Deferred Correction - local approach

A process of deferred correction based on Baker[1977] which we call a 'local' approach is now applied to (4.4.9). For the integral $\int_0^{rh} u(t)dt$, Gregory's formula may be written as

$$(4.4.10) \qquad \int_0^{rh} u(t)dt = T + h\sum_{s=1}^{p} c_s[\nabla^s u(rh) + (-1)^s \Delta^s u(0)],$$

where T represents the trapezoid rule approximation with stepsize h, and the other expression on the right hand side is the Gregory correction analogous to (4.1.5).

Using the above equation (4.4.10), we write the equations

$$\begin{aligned} & Y(0) = f(0) \\ & -\tfrac{1}{2}hk(h,0)Y(0) + [1 - \tfrac{1}{2}hk(h,h)]Y(h) = f(h) + C_1 \\ (4.4.11) \qquad & \\ & -\tfrac{1}{2}hk(2h,0)Y(0) - hk(2h,h)Y(h) \\ & \qquad + [1 - \tfrac{1}{2}hk(2h,2h)]Y(2h) = f(2h) + C_2 \\ & \qquad \ldots\ldots\ldots\ldots, \end{aligned}$$

where C_r, $r = 1(1)N$ represents Gregory correction in (4.4.10) with $u(t) = k(rh,t)Y(t)$.

In the rth stage of the local approach of deferred correction, we calculate Y(rh) in terms of previously accepted approximations to Y(0), Y(1), ..., $Y(\overline{r-1}h)$. The rth stage is initiated by computing $Y^{(o)}(rh)$ from the following formula in which $C_r^{(o)}$ is taken as zero.

$$(4.4.12)\quad [1 - \tfrac{1}{2}hk(rh,rh)]Y^{(o)}(rh) = f(rh) + h \sum_{j=1}^{r-1} k(rh,jh)Y(jh) + \tfrac{1}{2}hk(rh,0)Y(0) + C_r^{(o)}$$

The finite differences of the values k(rh, jh)Y(jh), $j = 0(1)r-1$, and $k(rh,rh)Y^{(o)}$ can be used in the Gregory correction to the trapezoid rule in order to obtain a correction term $C_r^{(o)}$. We then find a corrected value $Y^{(1)}(rh)$ satisfying

$$(4.4.13)\quad Y^{(1)}(rh) = Y^{(o)}(rh) + C_r^{(o)}/[1 - \tfrac{1}{2}hk(rh,rh)].$$

The process can be repeated, using $Y^{(1)}(rh)$ to compute $C_r^{(1)}$ and thereby $Y^{(2)}(rh)$, and so on. The process may be continued for a fixed number s of iterations or until $Y^{(s)}(rh)$, for some s, agrees to the required accuracy with $Y^{(s-1)}(rh)$. When $Y^{(s)}(rh)$ is accepted it is taken as Y(rh) in the formula used for calculating the subsequent values. The above method of IDC involves a local correction since Y(rh) is corrected before Y[(r+1)h] is.

Deferred Correction - Global Approach

We now introduce a global approach of deferred correction. In this approach we first find the solution $Y(rh) = Y^{(o)}(rh)$, $r = 0(1)N$ by the quadrature method (4.4.11) and then compute all the correction terms $C_r^{(o)}$, $r = 1(1)N$ using the expression for Gregory correction. Then we compute all the corrected values $Y^{(1)}(rh)$, $r = 1(1)N$ from (4.4.13).

Using $Y^{(1)}(rh)$ we compute $C_r^{(1)}$ and thereby $Y^{(2)}(rh)$, and the process is repeated adopting a stopping criterion similar to the one employed in the local IDC.

This approach is called 'global', because correction is made to all values of Y simultaneously.

Numerical results

The two approaches of IDC described above have been implemented together with the iterated deferred defect correction methods based on these approaches. The test runs have been performed in single precision (9-11 decimals) on the ICL 1904S computer at BHU, Varanasi. We consider again the test problem given in Section 2.4. More test results are included in Chapter 8.

In the test runs, we took the number of correction terms p of equation (4.4.10) as 1 to 4, and the number of iterations s as 1, 2 or 3. The basic interval chosen was

Table 4.4.10 Results for local IDC and IDDeC

p	N,m	IDC	IDDeC iteration number 1	2	3	4	5
1	4	1.1E-1	5.8E-3	5.9E-4	7.9E-4*		
	8	1.7E-2	7.9E-4	5.1E-5	3.9E-5	2.1E-7	1.9E-8
	12	5.4E-3	2.5E-4	1.6E-5	1.0E-6	6.4E-8	4.3E-9
	16	2.3E-3	1.0E-4	3.7E-6	3.8E-6	9.2E-6	5.0E-7
2	4	3.5E-2	1.3E-3	6.1E-4	5.9E-4*		
	8	4.4E-3	1.2E-4	2.0E-5	3.6E-6	6.2E-7	1.2E-7
	12	1.2E-3	3.5E-5	9.7E-6	2.7E-6	7.8E-7	2.2E-7
	16	5.1E-4	1.3E-5	6.8E-5	4.0E-6	2.1E-6	8.3E-7
3	4	2.9E-2	5.3E-4	3.0E-4*			
	8	4.7E-3	2.3E-4	2.0E-5	1.8E-6	1.5E-7	1.1E-8
	12	1.4E-3	1.1E-4	3.6E-5	1.1E-5	3.6E-6	1.1E-6
	16	5.0E-4	5.3E-5	1.8E-5	8.8E-6	3.4E-6	1.1E-6
[illegible]	8	4.7E-3	3.0E-4	4.4E-5	6.2E-6	8.7E-7	1.3E-7
	12	1.4E-3	2.4E-4	1.7E-4	1.1E-4	7.9E-5	5.4E-5
	16	6.3E-4	1.9E-4	3.9E-4	7.4E-4	1.7E-3	3.3E-3

Table 4.4.11 Results for global IDC and IDDeC

p	N,m	IDC	IDDeC iteration number				
			1	2	3	4	5
1	4	2.1E-1	9.1E-3	8.7E-4	7.8E-4*		
	8	8.0E-1	1.5E-3	2.4E-5	8.9E-7	3.8E-8	7.6E-9
	12	4.0E-2	4.3E-4	3.1E-6	2.0E-7	4.5E-9	8.1E-10
	16	2.4E-2	1.6E-4	9.8E-6	4.1E-6	8.7E-7	6.9E-7
2	4	4.1E-1	3.9E-2	4.3E-3	8.9E-4	6.3E-4	6.0E-4
	8	1.1E-1	5.0E-3	8.0E-4	1.9E-4	4.5E-5	1.1E-5
	12	5.3E-2	1.7E-3	4.8E-4	1.6E-4	5.6E-5	1.9E-5
	16	3.0E-2	7.6E-4	4.2E-4	2.7E-4	1.4E-4	6.3E-5
3		[illegible]	2.7E-2	2.3E-2	5.3E-5	2.7E-4	3.0E-4
	8	1.1E-1	2.6E-3	2.2E-4	1.1E-5	5.3E-7	3.0E-8
	12	[illegible]	3.1E-3	4.9E-4	5.7E-5	6.4E-6	7.7E-7
	16	[illegible]	2.1E-3	4.1E-4	1.4E-4	6.8E-5	6.6E-6
4	8	[illegible]	7.8E-3	1.6E-3	3.6E-4	7.7E-5	1.6E-5
	12	[illegible]	7.3E-3	5.4E-3	3.9E-3	2.8E-3	2.0E-5
	16	[illegible]	7.0E-3	1.4E-2	2.9E-2	5.0E-2	9.7E-2

Table 4.4.11 Results for global IDC and IDDeC

			IDDeC iteration number				
p	N,m	IDC	1	2	3	4	5
1	4	2.1E-1	9.1E-3	8.7E-4	7.8E-4*		
	8	8.0E-1	1.5E-3	2.4E-5	8.9E-7	3.8E-8	7.6E-9
	12	4.0E-2	4.3E-4	3.1E-6	2.0E-7	4.5E-9	8.1E-10
	16	2.4E-2	1.6E-4	9.8E-6	4.1E-6	8.7E-7	6.9E-7
2	4	4.1E-1	3.9E-2	4.3E-3	8.9E-4	6.3E-4	6.0E-4
	8	1.1E-1	5.0E-3	8.0E-4	1.9E-4	4.5E-5	1.1E-5
	12	5.3E-2	1.7E-3	4.8E-4	1.6E-4	5.6E-5	1.9E-5
	16	3.0E-2	7.6E-4	4.2E-4	2.7E-4	1.4E-4	6.3E-5
3		[illegible]	2.7E-2	2.3E-2	5.3E-5	2.7E-4	3.0E-4
	8	1.1E-1	2.6E-3	2.2E-4	1.1E-5	5.3E-7	3.0E-8
	12	[illegible]	3.1E-3	4.9E-4	5.7E-5	6.4E-6	7.7E-7
	16	[illegible]	2.1E-3	4.1E-4	1.4E-4	6.8E-5	6.6E-6
4	[illegible]	[illegible]	7.8E-3	1.6E-3	3.6E-4	7.7E-5	1.6E-5
	12	[illegible]	7.5E-3	5.4E-3	3.9E-3	2.8E-3	2.0E-5
	16	[illegible]	7.0E-3	1.4E-2	2.9E-2	5.0E-2	9.7E-2

[0,1] and the values for N were 4, 8, 12 and 16. For the IDDeC, the degree m was taken as N. Maximum absolute errors by the methods were noted. We found that the IDCs with s = 2 and 3 failed to improve the results for the IDC with s = 1 (the meaning of s as in the paragraph below the equation (4.4.13)).

In Tables 4.4.10 and 4.4.11 we display the errors produced by the methods based on the local and global versions of IDC with s = 1.

Our results show that deferred correction methods based on the local approach are preferable to those based on the global approach. They show also that the choice p = 4 produces worse results than p = 3 and that the choice of high values such as 16 for m should be avoided.

4.4.3 Nonlinear Volterra Equations

In this section we describe two deferred correction techniques applicable to the nonlinear Volterra integral equatio

$$(4.4.14) \qquad y(x) - \int_0^x k(x,t,y(t))dt = f(x), \qquad 0 \leq x \leq a.$$

With the use of Gregory's formula (4.4.10) we obtain the approximate equation

$$(4.4.15) \qquad Y_r - h \sum_{j=1}^{r-1} k(rh,jh,Y_j) - \tfrac{1}{2}\, hk(rh,0,Y_o)$$

$$- \tfrac{1}{2}\, hk(rh,rh,Y_r) = f_r + C_{r,p},$$

where p is a chosen fixed constant and

$$(4.4.16) \quad C_{r,p} = h \sum_{s=1}^{p} c_s [\, \Delta^s k(rh,0,Y_o) + (-1)^s \nabla^s k(rh,rh,Y_r)\,]$$

is the correction with the values of c_s given in Section 4.1.

There are two versions of deferred correction. One is the 'local' IDC which has been described by Kershaw[1974], and the other version is the 'global' IDC which is introduced below.

The local IDC

We begin by finding the solution $Y_1^{(o)}$ of equation (4.4.15) with r = 1 and $C_{r,p} = 0$.

In the rth stage $(2 \leq r \leq N)$ of the local IDC, we set the iteration count s to zero and perform the following four steps as many times as are necessary.

Step 1: With the use of values $Y_0^{(o)}$, $Y_1^{(o)}$, $Y_2^{(s)}$, ..., $Y_r^{(s)}$ and equation (4.4.16), compute $C_{r,p}^{(s)}$.

Step 2: Set $Y_r^{new} = Y_r^{(s)} + C_{r,p}^{(s)}$

Step 3: With the aid of the values $Y_0^{(o)}$, $Y_1^{(o)}$, $Y_2^{(s)}$, ..., $Y_{r-1}^{(s)}$ and the new value, Y_r^{new}, obtain $Y_r^{(s+1)}$ from (4.4.15)

Step 4: Set $s = s + 1$

For a given r, the sequence of iterates $Y_r^{(o)}$, $Y_r^{(1)}$,... obtained by the above process is not guaranteed to converge, and so $Y_r^{(p)}$ is accepted as the final approximation to $y(x_r)$. The accepted value is then denoted by $Y_r^{(o)}$.

The global IDC

In the global IDC, correction is done only after the solution is found at all the grid points by the basic discretization method. The correction steps for this IDC are the same as those of the local IDC except that now each of Steps 1 to 3 is performed for all $r = 2(1)N$. For $i = 1$, 2 and 3 let us therefore define Step i' as "For $r = 2(1)N$ do Step i of the local IDC".

In the global IDC, we begin by finding the solution $Y_r^{(o)}$, $r = 0(1)N$ of equation (4.4.15) with $C_{r,p} = 0$; this is the same as using the trapezoid method for (4.4.14).

Then, for $s = 0, 1, 2, \ldots$ do the three steps Step 1' to 3'.

Since the sequence of vectors $\bar{Y}^{(o)}$, $\bar{Y}^{(1)}$, $\bar{Y}^{(2)}$, ... produced by the above IDC may not converge, we take $\bar{Y}^{(p)}$ as the final approximation, and rename it as $\bar{Y}^{(o)}$.

Numerical Results

We consider the numerical solution of six problems by the local and global approaches of IDC and the corresponding IDDeC methods. The first five problems are Problems 1 - 5 of Section 2.3.2 and the sixth problem is the one given in Section 2.4. The range of integration was taken as [0,1], and the maximum absolute errors are tabulated in Tables 4.4.12 to 4.4.19.

In the tables we have encountered so far in this thesis, the degree m of polynomials used in the defect correction procedure has only been even and the results obtained have been highly accurate. However we may allow m to take odd numbers also in order to obtain equally good results. Table 4.4.12 illustrates the good performance of a global IDC based IDDeC with odd values for m.

In Table 4.4.13 we have listed the results of the global IDC with p = 1 and the corresponding IDDeC with m = 6.

We contrast the performances of the global and local IDC approaches and compare those of the corresponding IDDeC methods in Tables 4.4.14a and 4.4.14b. The local IDC yields more accurate results than the global IDC. But the IDDeC based on the global IDC produces results almost as good as the IDDeC based on the local IDC. It is interesting to observe that each successive iterations of IDDeC improve the previously obtained results in spite of the failure of the iterates of the underlying global IDC to converge.

Table 4.4.12 Global IDC with p = 2 and IDDeC with odd degree polynomials

Problem	m	N	IDC	IDC	IDDeC iteration number		
			(2,0)	(2,2)	1	2	3
1	5	5	2.2E-2	1.2E-2	6.7E-5	1.2E-6	4.7E-11
		10	5.1E-3	2.9E-3	8.2E-7	9.8E-10	1.4E-13
	7	7	1.1E-2	6.1E-3	1.4E-5	2.2E-7	1.1E-12
		14	2.6E-3	1.4E-3	1.0E-7	1.5E-9	1.5E-13
2	5	5	7.2E-3	7.2E-3	2.9E-5*		
		10	1.8E-3	1.9E-3	8.0E-7*		
	7	7	3.7E-3	3.8E-3	6.9E-7	6.7E-7*	
		14	9.2E-4	1.0E-3	5.7E-9	5.4E-9*	
4	5	5	5.3E-3	2.9E-3	1.1E-6	1.1E-8	1.7E-8
		10	1.3E-3	6.7E-4	5.3E-8	1.9E-10	3.1E-10
	7	7	2.7E-3	1.4E-3	4-0E-7	4.0E-9	9.7E-11
		14	6.9E-4	3.3E-4	1.7E-8	6.6E-11	4.3E-13

Table 4.4.13 Global IDC with p = 1 and IDDeC with m = 6

		IDC	IDDeC iteration number				
Problem	N	(1,1)	1	2	3	4	5
1	6	6.2E-3	2.0E-4	1.2E-5	7.5E-7	4.7E-8	3.0E-9
	12	1.2E-3	1.0E-5	1.8E-7	3.4E-9	6.9E-11	3.2E-13
	24	2.6E-4	4.9E-7	1.9E-9	9.3E-12	1.5E-13	8.9E-14
2	6	8.1E-4	4.2E-6	4.0E-6*			
	12	7.1E-5	2.7E-8	2.5E-8*			
	24	4.1E-6	4.2E-10	2.2E-10*			
3	6	1.2E-3	6.3E-6	1.1E-7	9.8E-10	2.9E-9*	
	12	2.5E-4	2.4E-7	5.6E-10	1.1E-11	1.0E-11*	
	24	5.6E-5	10.0E-9	3.4E-12	1.2E-13	1.2E-13*	
4	6	1.4E-3	1.8E-5	4.7E-7	9.9E-9	6.6E-10	7.2E-10
	12	3.1E-4	1.2E-6	1.8E-8	1.9E-10	4.5E-12	5.3E-12
	24	7.0E-5	8.5E-8	6.1E-10	3.6E-12	1.1E-13	6.6E-14
5	6	2.4E-3	1.4E-4	2.4E-6	3.1E-8	4.4E-10	7.0E-12
	12	9.1E-4	8.8E-6	3.3E-8	1.7E-10	4.5E-13	3.4E-15
	24	2.7E-4	5.3E-7	5.1E-10	7.9E-13	1.4E-14	2.2E-16

Table 4.4.14a Local IDC with p = 4 and IDDeC with m = N = 12

Problem	IDC				IDDeC			
	(4,0)	(4,1)	(4,2)	(4,4)	(4,1)	(4,2)	(4,4)	(4,5)
1	3.5E-3	5.0E-5	3.3E-4	3.5E-4	3.0E-7	3.3E-9	9.0E-14	8.7E-16
2	1.2E-3	6.4E-6	1.6E-5	1.6E-5	2.5E-9	4.6E-11	3.4E-11*	
3	7.7E-4	4.2E-6	2.8E-5	2.9E-5	1.1E-8	6.4E-11	2.1E-15	2.0E-17
4	9.4E-4	4.0E-6	2.8E-5	2.8E-5	1.1E-7	1.2E-8	9.9E-12	1.3E-12
5	1.6E-3	1.4E-6	1.4E-6	1.4E-6	7.4E-7	3.0E-8	1.3E-10	1.3E-12

Table 4.4.14b Global IDC with $p = 4$ and IDDeC with $m = N = 12$

Problem	IDC				IDDeC			
	(4,0)	(4,1)	(4,2)	(4,4)	(4,1)	(4,2)	(4,4)	(4,5)
1	3.5E-3	9.2E-4	2.0E-3	2.9E-3	3.1E-7	3.3E-9	8.9E-14	4.6E-15
2	1.2E-3	7.8E-5	1.4E-3	1.3E-3	2.5E-9	4.6E-11	3.4E-11*	
3	7.7E-4	1.9E-4	4.1E-4	6.0E-4	1.1E-8	6.4E-11	2.1E-15	1.2E-16
4	9.4E-4	2.5E-4	4.6E-4	7.1E-4	1.1E-7	1.2E-8	9.9E-12	1.3E-12
5	1.6E-3	1.2E-3	3.9E-3	2.8E-3	7.4E-7	3.0E-8	1.3E-10	1.3E-12

Table 4.4.15 displays the results of IDDeC with m = 8 and the underlying local IDC method with a single deferred correction involving upto fourth order differences in Gregory's formula.

Table 4.4.16 shows the power of the IDDeC method in producing highly accurate solutions for relatively large stepsizes.

Table 4.4.17 reveals the fact that for Problems 2 and 3, increase in the degree of polynomials gives rise to an increase in accuracy of results by the IDDeC method.

Tables 4.4.18 and 4.4.19 show that the performance of IDDeC based on IDC(p,p) for p = 1, 2, 3 and 4 generally becomes better as p increases.

For practical purposes for the numerical solution of Volterra equations, we prefer IDDeC method based on the local IDC to that based on the global IDC. Moreover, we choose the parameters m, p and s of the IDDeC method such that m = 6 or 8, p = 3 or 4 and s = 1 or 2 only.

Table 4.4.15 IDDeC with m = 8 and the underlying local IDC (4,1) method

		IDC		IDDeC iteration		
Problem	N	(4,0)	(4,1)	1	3	5
1	8	8.2E-3	1E-4	4E-7	3E-11	7E-13
	16	2.0E-3	2E-5	2E-8	7E-13	2E-13
2	8	2.8E-3	2E-5	6E-8	9E-8*	
	16	7.0E-4	3E-6	2E-10	2E-10*	
3	8	1.7E-3	1E-4	1E-7	1E-11	4E-12
	16	4.3E-4	2E-6	8E-10	5E-14	1E-14
4	8	2.1E-3	1E-5	8E-7	9E-10	4E-12
	16	5.3E-4	2E-6	2E-8	7E-12	8E-14
5	8	3.6E-3	1E-5	2E-6	5E-9	8E-12
	16	8.9E-4	3E-7	5E-8	4E-12	5E-14

Table 4.4.16 Performance of IDDeC for small values of m = N based on local IDC (2,2)

		iteration number		
Problem	N	1	3	5
1	4	2.3E-4	3.5E-8	9.7E-12
	6	4.9E-5	1.5E-8	4.0E-12
2	4	1.6E-4*		
	6	4.0E-6*		
3	4	2.9E-6	1.8E-6*	
	6	5.3E-8	2.9E-9*	
4	4	3.6E-6	1.6E-7*	
	6	8.2E-7	7.2E-10*	
5	4	2.4E-6	9.4E-10	3.2E-13
	6	5.5E-7	1.9E-10	5.9E-14

Table 4.4.17 Local IDC with $p = 4$ and IDDeC$_m$ with $m = 4$, 6 and 12

Problem	N	IDC	IDDeC$_4$		IDDeC$_6$		IDDeC$_{12}$	
		(4,4)	(4,1)	(4,2)	(4,1)	(4,2)	(4,1)	(4,2)
2	12	2E-5	5E-8*		2E-8*		2E-9	3E-11
	24	2E-6	1E-8*		2E-10*		5E-12	1E-13
3	12	3E-6	2E-10	3E-10*	1E-10	1E-12	1E-8	3E-13
	24	4E-7	3E-12	4E-11*	5E-11	1E-13	4E-11	9E-14

Table 4.4.18 Results for problem 1 by IDDeC with m = 6 based on local IDC with p = 1, 2, 3 and 4

p	N	IDC (p,p)	IDDeC		
			(p,1)	(p,2)	(p,3)
1	24	5E-5	6E-8	2E-10	1E-12
	48	6E-6	1E-9	8E-13	3E-14
2	24	4E-5	9E-9	1E-11	9E-14
	48	5E-6	2E-10	4E-14	1E-15
3	24	4E-5	7E-9	2E-11	9E-14
	48	5E-6	2E-10	7E-14	7E-16
4	24	4E-5	1E-8	9E-12	2E-13
	48	5E-6	3E-10	8E-14	5E-16

Table 4.4.19 Results for problem 6 by IDDeC with $m = 6$ based on local IDC with $p = 1, 2, 3, 4$ and $N = 12$

~~Problem~~ p	IDC (p,p)	IDDeC (p,1)	IDDeC (p,3)
1	8.0E-3	1.4E-4	2.1E-7
2	3.7E-3	8.1E-5	2.1E-8
3	4.2E-3	4.2E-6	1.4E-8
4	4.2E-3	1.9E-6	1.3E-8

4.5 Conclusion

We have presented the method of IDDeC which is another new iterative technique for improving the accuracy of numerical solution of operator equations; we used the Fredholm integral equations to explain the ideas about the method and analyze its asymptotic behaviour.

The IDDeC method is a generalization of the IDC and IDeC methods. It has a faster convergence than its constituent methods, but is free from their disadvantages. For instance, IDDeC removes the uncertainty in convergence of the iterates of IDC, and it is less expensive than IDeC in that the number of applications of the defect correction to obtain a given order of accuracy by the IDDeC is less than that required by the IDeC on which the former method is based.

We have applied IDDeC based on the IDC methods that use quadrature rules for FIEs and VIEs. In the case of linear FIEs, we have considered the underlying IDC methods that are based on Simpson's as well as the trapezoid rule with correction. But in the case of linear and nonlinear VIEs, we have described two approaches of IDC based on Gregory's formula.

Now we shall remark on the computational experiments on IDDeC methods. When applied to FIEs, IDDeC based on Gregory's formula produced confirmatory results for our theoretical analysis on the order of convergence. It was also found that the above method is easier to implement and is more desirable than the IDDeC based on Simpson's rule with correction. As for the application to linear VIEs, the IDDeC method based on the local IDC performs better than the one based on the global IDC. However, in the case of nonlinear VIEs, the above two IDDeC methods produce equally good results in spite of the superiority of the local IDC over the global IDC.

CHAPTER 5

THE METHOD OF SUCCESSIVE EXTRAPOLATED ITERATED DEFERRED DEFECT CORRECTION

5.1 Introduction

In Chapter 4 we introduced the method of Iterated Deferred Defect Correction (IDDeC) which is the result of applying the method of Iterated Defect Correction (IDeC) on the technique of Iterated Deferred Correction (IDC). We also found that when applied to Fredholm integral equations of the second kind, the IDDeC method based on the IDC that uses Gregory's formula possesses an asymptotic expansion in even powers of the stepsize h.

In the next section we introduce a new method which is a judicious combination of the three standard techniques for iterative improvement, namely deferred correction, defect correction and Richardson extrapolation.

5.2 Successive Extrapolated IDDeC (SEIDDeC)

We apply Richardson extrapolation on the method of ILDeC, and name the resulting technique as the method of Successive Extrapolated Iterated Deferred Defect Correction (SEIDDeC).

We now present this new method SEIDDeC as applied to Fredholm equations of the second kind. Start with a basic stepsize of $h = (b-a)/N$. Find the numerical solution of (2.2.1) by the ID$\mathbb{C}$(p,p) method that uses the trapezoid rule with Gregory correction consisting of differences of order not greater than p and involves p iterations of deferred correction. Then apply the method of IDeC, with a fixed degree m of polynomials, on the above method of IDC. Denote by $Y(x,h)$ the solution thus obtained.

The order q of accuracy in this solution is given by

$$(5.2.1) \qquad q = \min[(s+1)(p+2),\ m+2],$$

where s is the number of times defect correction has been performed (see Theorem 4.2 in Section 4.3.2). For a suitable choice of s, the solution $Y(x,h)$ is of order m+2 and an asymptotic expansion for the global error $Y(x,h) - y(x)$ is given by

$$(5.2.2) \quad Y(x,h) - y(x) = C_{m+2}(x)h^{m+2} + C_{m+4}(x)h^{m+4} + \ldots$$

Now apply Richardson extrapolation as in Section 3.3. Continue the extrapolation process until convergence, to within a user-prescribed tolerance, of the extrapolated values is obtained. If this is not achieved in a few iterations, say 2, the entire procedure is repeated with a smaller basic stepsize.

The asymptotic expansion for the method SEIDDeC is given by

$$(5.2.3) \quad Y^j(x,h) - y(x) = C_{m+2+2j}(x)h^{m+2+2j} + C_{m+4+2j}(x)h^{m+4+2j} + \ldots\ldots,$$

where j is the order of extrapolation. This is analogous to equation (3.3.3) which gives the asymptotic expansion for the SEIDeC method.

The SEIDDeC method described above is denoted by $SEIDDeC_m(p,s)$. We observe that SEIDDeC is to IDDeC what SEIDeC is to IDeC.

5.3 Application to Fredholm Equations

In practical situations we start with an IDC for which p is 2 or 3, and then apply IDeC for which m is 6 or 8, and the number of iterations s is 1. In fact we can choose $2(p+2) > m+2$ or

$$(5.3.1) \qquad p > m/2 - 1.$$

This ensures that the IDDeC(p,1) will be of order m+2 (see equation (4.3.19a)). Furthermore, Richardson extrapolation may then be applied on this IDDeC. The method described in the above paragraph is $SEIDDeC_m(p,1)$ for which p and m satisfy the relation (5.3.1). Since the defect correction procedure is costlier than deferred correction, we always try to minimize the number s of iterations to one.

For our numerical experiments, we considered IDDeC(p,1) methods with m = 4 and 6 based on IDC(p,p) with p = 2 and 3 respectively and the corresponding $\text{SEIDDeC}_m(p,1)$ methods. The results for IDDeC and SEIDDeC correspond to those of IDeC and SEIDDeC (see Tables 3.4.1 and 3.4.2) and they are identical except for the entries which have reached the machine accuracy. Hence we do not display here the computational results for IDDeC and SEIDDeC.

5.4 Conclusion

In this chapter we have presented the SEIDDeC technique which is a generalization of the methods of deferred correction, defect correction and Richardson extrapolation. The order of convergence of this method is m+2+2j, where m is the degree of polynomials used in the defect correction procedure, and j is the order of extrapolation.

An advantage of the method of SEIDDeC over the method of SEIDeC is that the former usually requires at most one iteration of defect correction whereas the latter requires m/n iterations, n being the order of the basic discretization scheme.

The SEIDDeC method is applicable to any basic discretization scheme for the numerical solution of an operator equation whenever the scheme admits an asymptotic expansion for the IDDeC method on which the SEIDDeC is based.

CHAPTER 6

THE METHOD OF ITERATED DEFECT CORRECTION ON EXTRAPOLATION

6.1 Introduction

In Chapter 3 we applied Richardson extrapolation on the method of IDeC and thereby introduced the method of Successive Extrapolated IDeC (SEIDeC). In this chapter we combine the methods of extrapolation and IDeC in the reverse order, by performing IDeC on the method of Richardson extrapolation. The resulting technique is named as the method of Iterated Defect Correction on Extrapolation. This new method is presented as applied to the Fredholm integral equation

$$(6.1.1) \qquad y(x) - \int_a^b k(x,t)\, y(t)\, dt = f(x), \qquad a \leq x \leq b.$$

For the above FIE, quadrature methods based on Newton-Cotes quadrature formulae were discussed in Section 2.2, and the principle of Richardson extrapolation was explained in Section 3.3. The method of IDeC on Extrapolation is introduced in the next section, and its asymptotic expansion is derived in Section 6.3.

6.2 Iterated Defect Correction on Extrapolation (IDeCE)

Considering extrapolation of the first order, we now present the method of Iterated Defect Correction on Extrapolation (IDeCE) as applied to a quadrature method Quad(h) with a stepsize h for the numerical solution of (6.1.1).

We first solve (6.1.1) by the methods Quad(h) and Quad($\frac{h}{2}$), and then obtain extrapolated values $Z_i^{(0)}$ at the points $x_i = a + ih$, $i = 0(1)N$ by Richardson extrapolation of order one.

As in Section 2.2.1, we proceed to apply the method of IDeC by constructing a polynomial function $P_m^{(o)}(x)$ which interpolates the points $(x_i, Z_i^{(o)})$, $i = 0(1)N$. We then solve the NP

$$(6.2.1) \qquad y(x) - \int_a^b k(x,t)\, y(t)\, dt = P_m^{(o)}(x) - \int_a^b k(x,t) P_m^{(o)}(t)\, dt$$

first by Quad(h) and then by Quad($\frac{h}{2}$), and finally obtain the resulting extrapolated values $\gamma_i^{(o)}$ at x_i, $i = 0(1)N$.

We then compute an improved solution $Z_i^{(1)}$ with the use of Zadunaisky's principle

$$(6.2.2) \qquad Z_i^{(1)} = Z_i^{(o)} + P_m^{(o)}(x_i) - \gamma_i^{(o)}$$

The procedure above is a simple defect correction on extrapolation.

Repeated application of defect correction leads to the method of Iterated Defect Correction on Extrapolation (IDeCE).

Practical difficulties

There are two difficulties that are encountered in the implementation of the IDeCE method discussed above. One difficulty is caused by the nonavailability of extrapolated values at the midpoints of the original gridpoints. This difficulty arises when finding the solution of NP by Quad$(\frac{h}{2})$ which calls for a knowledge of the values of the function on the right hand side of (6.2.1) at all the $x_i' = a + i\frac{h}{2}$, $i = 0(1)2N$. Besides, we wish the values $P_m^{(o)}(x_i')$ at the above points x_i' to have the same order of accuracy as the extrapolated values available at the original grid points. To avoid this difficulty, we first find ${}^{mid}Z_j^{(o)}$ at x_j' , $j = 1(2)2N-1$, using the equation

$$(6.2.3) \qquad {}^{mid}Z_j = \sum_{i=0}^{N} w_i k(x_j, t_i) Z_i^{(o)} + f(x_j)$$

which is obtained from (6.1.1); we then construct $P_m^{(o)}(x)$ interpolating the 2N+1 points $(x_i, Z_i^{(o)})$, $i = 0(1)N$ and $(x_j', {}^{mid}Z_j^{(o)})$, $j = 1(2)2N-1$.

Another difficulty is that the IDeCE requires extra storage. For an efficient computation of the solutions to OP and NP by a quadrature method, the LU decomposition of the matrix of coefficients of the system of linear equations resulting from the OP has to be saved. During any defect correction process, IDeCE thus requires a separate storage for a $(2N+1) \times (2N+1)$ coefficient matrix in connection with the use of Quad$(\frac{h}{2})$ in addition to the storage needed for an $(N+1) \times (N+1)$ matrix for the use of Quad(h). This is not so in the case of the SEIDeC method.

6.3 Asymptotic Error Expansion

We now give a derivation for an asymptotic expansion for the global error in the solution to (6.1.1) obtained by the IDeCE method described in the last section. Since this derivation is similar to that for the asymptotic expansion for the IDeC method discussed in Section 3.2, the details will be omitted.

We assume that m is an even integer greater than two. Let the errors in the solutions of CP (6.1.1) by the methods Quad$_T(h)$ and Quad$_T(\frac{h}{2})$ based on the trapezoid rule, and the method of extrapolation be denoted by $\tau_{i,h}$, $\tau_{i,\frac{h}{2}}$ and $\tau_{i,h}^{(1)}$ respectively, and those of NP (6.2.1) by $\varepsilon_{i,h}$, $\varepsilon_{i,\frac{h}{2}}$ and $\varepsilon_{i,h}^{(1)}$. Then for the points $x_i = a + ih$ we have

$$\tau_{i,h} = A_2(x_i)h^2 + A_4(x_i)h^4 + \ldots,$$

$$\tau_{i,\frac{h}{2}} = A_2(x_i)\frac{h^2}{4} + A_4(x_i)\frac{h^4}{16} + \ldots, \text{ and}$$

(6.3.1) $$\tau^{(1)}_{i,h} = A^{(1)}_4(x_i)h^4 + A^{(1)}_6(x_i)h^6 + \ldots .$$

Consequently,

(6.3.2) $$Z^{(o)}_i - y(x_i) = B^{(1)}_4(x_i)h^4 + B^{(1)}_6(x_i)h^6 + \ldots .$$

Now the relation between the errors by the $\text{Quad}_T(h)$ method applied to OP and NP is given by

(6.3.3) $$\varepsilon_{i,h} = \tau_{i,h} + O(h^6).$$

Therefore,

(6.3.4) $$\varepsilon^{(1)}_{i,h} = A^{(1)}_4(x_i)h^4 + \alpha^{(1)}_6(x_i)h^6 + \ldots,$$

and analogous to (6.3.2) we have

(6.3.5) $$\gamma^{(o)}_i - P^{(o)}_m(x_i) = B^{(1)}_4(x_i)h^4 + \beta^{(1)}_6(x_i)h^6 + \ldots .$$

The use of Zadunaisky's principle (6.2.2) yields

(6.3.6) $$Z^{(1)}_i - y(x_i) = B^{(2)}_6(x_i)h^6 + B^{(2)}_8(x_i)h^8 + \ldots .$$

Moreover, the required error expansion for the IDeCE method based on $\text{Quad}_T(h)$ is

(6.3.7) $$Z^{(s)}_i - y(x_i) = B^{(s+1)}_q(x_i)h^q + B^{(s+1)}_{q+2}(x_i)h^{q+2} + \ldots .$$

where $q = \min[2(s+1)+2, m+2]$, and $s = 0, 1, 2, \ldots$.

In the case of the $\mathrm{Quad}_S(h)$ method based on Simpson's rule, the asymptotic expansion for the corresponding IDeCE method is given by (6.3.7) with $q = \min[4(s+1)+2, m+2]$.

6.4 Conclusion

In this chapter we have presented the method of IDeC on Extrapolation as applied to the quadrature methods for the numerical solution of Fredholm integral equations of the second kind. We have also derived asymptotic expansion for the above IDeCE method.

A common feature of the IDeCE and IDeC methods is that each defect correction increases the order of accuracy in their solutions by the order of the basic discretization scheme, and that the attainable order is limited by the degree of polynomials.

When the methods SEIDeC and IDeCE are compared, SEIDeC is found superior to IDeCE. The disadvantages of IDeCE are that it requires more work and more storage than SEIDeC, and that its order is limited.

To summarize, we find that the order in which the methods IDeC and Richardson extrapolation are combined does matter and that the order which is used in SEIDeC is to be preferred.

CHAPTER 7

THE METHOD OF SUCCESSIVE UPDATED ITERATED DEFECT CORRECTION

7.1 Introduction

In this chapter we present a new technique called the method of Successive Updated Iterated Defect Correction (SUIDeC). This technique is a variant of the relatively new IDeC method. The SUIDeC method makes use of interpolatory cubic splines instead of polynomials which have been employed in the defect correction procedures discussed in the previous chapters.

We introduce the new method as applied to the following nonlinear Volterra integral equation of the second kind:

$$(7.1.1) \qquad y(x) = f(x) + \int_0^x k(x,t,y(t))\,dt, \qquad 0 \leq x \leq a.$$

As in Section 2.3, we denote the unique solution of this equation by $y(x)$.

Let N be a positive even integer, $h = a/N$ and $x_r = t_r = rh$, $r = 0(1)N$. Let us represent an approximation to $y(x_r)$ by Y_r, and denote $f(x_r)$ by f_r and $k(x_r, t_j, Y_j)$ by $k_{r,j}$. Then Noble's method is given by the equations

$$(7.2.1) \qquad y(x) = f(x) + \int_0^x k(x,t,y(t))\,dt, \qquad 0 \leq x \leq x_i$$

We then compute a solution value $Y_i^{(1)}$ at x_i for this equation by Noble's method, and its iterative improvements $Y_i^{(s)}(s \geq 2)$ by the method of IDeC. We denote by Y_i the improved value obtained at the end of the final iteration of the IDeC.

The i-th stage

Having determined the values Y_j at x_j for $j = 0(1)i-1$, we use (2.3.4a) or (2.3.4b) of Noble's method as required, and find $Y_i^{(1)}$ as an approximation to $y(x_i)$. To improve the solution value just found, we estimate its unknown error $y(x_i) - Y_i^{(1)}$. We fit the points (x_j, Y_j), $j = 0(1)i-1$ and $(x_i, Y_i^{(1)})$ by an interpolatory cubic spline function $P_i^{(1)}(x)$, and construct the problem

$$(7.2.2) \qquad y(x) = P_i^{(1)}(x) - \int_0^x k(x,t,P_i^{(1)}(t))\,dt + \int_0^x k(x,t,y(t))\,dt, \qquad 0 \leq x \leq x_i.$$

Since $P_i^{(1)}(x) - \int_0^x k(x,t,P_i^{(1)}(t))\,dt - f(x)$ is small, (7.2.2) is a neighbouring problem (NP) of (7.2.1). The exact solution of this NP is the known function $P_i^{(1)}(x)$. Now, starting with updated values Y_j, $j = 0(1)i-1$, and using (2.3.4a) or (2.3.4b) of Noble's method as required, we find the solution $\eta_i^{(1)}$ of NP (7.2.2) at x_i. The error in this solution value is known, and is equal to

$$P_i^{(1)}(x_i) - \eta_i^{(1)}.$$

Since both (7.2.1) and (7.2.2) are solved by the same method, we postulate that the error $y(x_i) - Y_i^{(1)}$ is approximately equal to the known error $P_i^{(1)}(x_i) - \eta_i^{(1)}$. Therefore,

$$Y_i^{(2)} = Y_i^{(1)} + (P_i^{(1)}(x_i) - \eta_i^{(1)})$$

is a better approximation to $y(x_i)$ than $Y_i^{(1)}$.

The above technique may be repeated to compute improved approximations $Y_i^{(s+1)}$ for $s = 2, 3, \ldots,$ by performing the following steps:

(i) Construct the NP

(7.2.3) $$y(x) = P_i^{(s)}(x) - \int_0^x k(x,t,P_i^{(s)}(t))\,dt + \int_0^x (x,t,y(t))\,dt, \quad 0 \leq x \leq x_i,$$

where $P_i^{(s)}(x)$ is an interpolatory cubic spline function fitting the points (x_j, Y_j), $j = 0(1)i-1$, and $(x_i, Y_i^{(s)})$.

(ii) Find the solution value $\eta_i^{(s)}$ for NP, using the relevant equation of Noble's method.

(iii) Compute the improved approximation

$$(7.2.4) \qquad Y_i^{(s+1)} = Y_i^{(1)} + P_i^{(s)}(x_i) - \eta_i^{(s)}.$$

For a preassigned tolerance TOL, we denote by s_i the value of s for which $|Y_i^{(s+1)} - Y_i^{(s)}| < \text{TOL}$ is satisfied. We then take $Y_i^{(s_i+1)}$ as the updated solution value Y_i at x_i for (7.2.1).

The values Y_i, $i = 0(1)N$ produced by the method of SUIDeC form the numerical solution of (7.1.1).

Implementation remarks

In our implementation of SUIDeC, we calculate the end-derivates required for the cubic splines $P_i^{(s)}(x)$ of (7.2.3) by making use of at least four ordinates (see equation (2.11) in Tewarson[1980]). Hence we need three starting values Y_0, Y_1 and Y_2 for SUIDeC.

The integral $\int_0^x k(x,t,P_i^{(s)}(t))\,dt$ in (7.2.2) and (7.2.3) is evaluated using standard routines such as those provided by NAG.

Cost

We shall now find the major difference in the cost of the global IDeC method that uses splines and the SUIDeC method.

In the case of the global approach of IDeC using splines, an NP is of the form:

$$(7.2.5) \quad y(x) = P(x) - \int_0^x k(x,t,P(t))\,dt + \int_0^x k(x,t,y(t))\,dt, \qquad 0 \leq x \leq a$$

where P(x) is a cubic spline function interpolating the points (x_i, Y_i), $i = 0(1)N$ and the Y_i in turn constitute the most accurate solution available during the defect correction process. Thus during any iteration of the above IDeC only one spline function (through N+1 points) is to be computed. On the other hand, in the case of SUIDeC an NP is constructed for each subinterval $[0, x_i]$, $4 \leq i \leq N$, and this NP requires a spline through i+1 points. Thus for each defect correction the number of splines to be computed in SUIDeC is about N, whereas only a single spline is to be computed in the global IDeC method.

Apart from this difference, the costs of the methods IDeC and SUIDeC are essentially the same.

7.3 Application to Volterra Equations

We applied Noble's method, the global approach of IDeC using splines, and our method SUIDeC on four problems; the first three are Problems 1 - 3 of Section 2.3.3 (Netravali[1973] and Baker[1977]), and the remaining one is Problem 4, from Baker[1977], given below:

Problem 4

$$y(x) = 2 - e^{x} + \int_{0}^{x} e^{x-t} y^2(t)\, dt, \qquad 0 \leq x \leq 1$$

exact solution: $y(x) = 1$

Problems 1 to 4 were solved for $N = 16$ and 32 and the absolute values of the errors $e_n = y(x_n) - Y_n$, $n = 3(1)N$ were computed. In Table 7.3.1 we give these values corresponding to a few points in $[0, 1]$. The value chosen for the tolerance TOL in the SUIDeC method was 10^{-15}. The word 'iter' denotes the number of iterative improvements done to improve the solution values.

7.4 Conclusion

We have described a version of the method of Successive Updated Iterated Defect Correction as applied to (7.1.1).. We compared the results with those obtained by the method of IDeC involving the global approach and the use of cubic splines, and found that the local approach (SUIDeC) yields better results than the global.

Comparing the performances of the three methods, namely IDeC with polynomials discussed in Section 2.3, its analogous IDeC with splines, and SUIDeC of this chapter, we find that the first IDeC is the best and the cheapest.

Table 7.3.1 Absolute errors in the solutions by the methods of Noble, global IDeC and SUIDeC.

Problem	N	x	Noble	IDeC	iter	SUIDeC	iter
1	16	.5	.16(-8)	.50(-11)	3	.11(-10)	2
		1.0	.90(-6)	.18(-6)		.23(-7)	4
	32	.5	.98(-10)	.34(-12)	3	.75(-12)	2
		1.0	.55(-7)	.11(-7)		.16(-8)	3
2	16	.5	.90(-7)	.50(-8)	2	.36(-8)	3
		1.0	.11(-5)	.14(-6)		.57(-7)	3
	32	.5	.57(-8)	.38(-9)	3	.30(-9)	2
		1.0	.66(-7)	.97(-8)		.49(-8)	2
3	16	.5	.66(-7)	.36(-8)	3	.12(-8)	3
		1.0	.26(-6)	.73(-7)		.77(-8)	3
	32	.5	.41(-8)	.21(-9)	3	.64(-10)	2
		1.0	.16(-7)	.44(-8)		.45(-9)	2
4	16	.5	.11(-6)	.38(-7)	3	.67(-8)	3
		1.0	.63(-6)	.43(-6)		.53(-7)	3
	32	.5	.68(-8)	.23(-8)	3	.12(-8)	2
		1.0	.38(-7)	.26(-7)		.68(-8)	3

CHAPTER 8

RELATIVE PERFORMANCES OF METHODS OF ITERATIVE IMPROVEMENT

Various iterative improvement techniques for solving integral equations are compared in this chapter.

8.1 Applicability to Fredholm Integral Equations

In this section we compare the following methods of iterative improvement which are applicable to Fredholm integral equations: Extrapolation, IDC, IDeC, IDDeC, SEIDeC and SEIDDeC.

The methods IDeCE and SUIDeC are not in the above list because IDeCE has already been compared with SEIDeC (see Chapter 6), and SUIDeC is not applicable to FIEs.

We compare the methods with regard to the increase in order of convergence per improvement or correction, the sequence of orders of convergence of the iterates, and the maximum possible order of convergence. We assume that n is the order of the basic quadrature method on which techniques of iterative improvement are applied, and that, in the case of deferred correction methods, p is the order upto which differences are included as correction to the basic quadrature formula used. We summarize our findings in Table 8.1.

Table 8.1 Comparison of Iterative Methods of Improvement

method	increase in order	sequence of orders of iterates
Extrapolation	2	n+2, n+4, ...
IDC	1	n+1, n+2, ..., p+2.
IDeC	n	2n, 3n, ..., m+2.
IDDeC	p+2	2(p+2), 3(p+2), ..., m+2.
SEIDeC	2	m+4, m+6, ...
SEIDDeC	2	m+4, m+6, ...

As noted in the table, if the iterative methods for FIE(2.2.1) are based on the trapezoid rule, a single iteration of IDC increases the order of accuracy by one; that of IDeC, by two; and that of IDDeC, by p+2.

Each of our methods IDDeC, SEIDeC and SEIDDeC is superior to Lindberg's technique, which improves the order of its iterates by n only. Moreover, the results obtained by our methods are far better than those by his. (Contrast our results, for instance, in Table 4.4.4a with his in p. 79 of Lindberg[1976].)

We recall that IDDeC is based on an IDC which is of order p+2, and that SEIDeC and SEIDDeC are based respectively on an IDeC and IDDeC, each of which has the maximum order m+2. This order is attained after (m+2)/n-1 applications of defect correction in the case of SEIDeC, and after $(m+2)/(p+2) - 1$ applications in the case of SEIDDeC. Thus, for a suitable choice of p, IDDeC needs only one or two applications of the defect correction process to produce results with error $O(h^{m+2})$.

It is possible to modify our SEIDeC and SEIDDeC methods so that their underlying defect correction technique has a variable order of convergence, namely $(s+1)n$ or $(s+1)(p+2)$, instead of the constant order $m+2$.

For general purpose routines for solving FIEs, it seems best to use IDDeC or SEIDDeC based on the trapezoid rule.

8.2 Applicability to Volterra Integral Equations

In this section, we compare accleration techniques applicable to VIEs.

Now that the SUIDeC method of the previous chapter has already been found inferior to the IDeC method of Chapter 2, we confine our attention only to the three methods IDC, IDeC and IDDeC.

Using 9-11 significant digits, two problems in linear VIEs were solved by the three methods based on the trapezoid rule. Problem 1 is given below and Problem 2 is Problem 3 of Section 2.3.2.

Problem 1

$$y(x) + \int_0^x e^{x-t}\, y(t)\, dt = 2x\,, \qquad 0 \leq x \leq 1$$

$$\text{exact solution:} \quad y(x) = 2x - x^2.$$

The maximum absolute errors obtained are given in Table 8.2. The performance of IDDeC(2,1) is better than that of IDeC(2) and IDC. This is also true for the problem considered in Section 2.4 (see Tables 2.4.1 and 4.4.10).

In the case of linear and nonlinear VIEs, our computational results show that, to produce solutions of a given accuracy, IDDeC generally requires fewer applications of defect correction than the corresponding IDeC. Thus it is cheaper to use IDDeC than IDeC.

8.3 Final Remarks

The method is IDDeC is a promising method with applicability to differential as well as integral equations. In the case of linear FIEs with smooth kernels and functions, SEIDDeC surpasses IDDeC.

The hierarchical order of the methods as regards their capability to produce highly accurate solutions is as IDDeC, IDeC and IDC.

Table 8.2 Maximum Errors by IDeC, Local IDC, and IDDeC

	N	IDeC with m = N			IDC	IDDeC with m = N		
		1	2	3	(2,2)	(2,1)	(2,2)	(2,3)
Problem 1	4	7.9E-4	8.6E-5	6.7E-6	1.8E-3	6.8E-5	3.2E-6	1.6E-7
	8	8.2E-5	6.6E-6	4.5E-7	1.6E-4	6.4E-6	8.8E-7	1.2E-7
	12	2.2E-5	1.5E-6	9.1E-8	3.8E-5	1.5E-6	3.1E-7	6.4E-8
	16	8.6E-6	1.1E-6	1.1E-6	1.6E-5	5.8E-7	2.5E-7	9.3E-8
Problem 2	4	1.6E-4	7.8E-6	1.5E-6	5.5E-5	3.0E-6	1.8E-6*	
	8	9.5E-6	9.3E-8	1.6E-9	1.1E-5	1.7E-8	2.0E-10	3.6E-11
	12	1.8E-6	2.3E-9	9.5E-11	4.4E-6	1.2E-9	4.4E-10	3.6E-11
	16	5.3E-7	1.6E-8	1.7E-8	2.1E-6	1.1E-9	3.5E-10	4.6E-10

Table 8.2 Maximum Errors by IDeC, Local IDC, and IDDeC

	N	IDeC with m = N			IDC	IDDeC with m = N		
		1	2	3	(2,2)	(2,1)	(2,2)	(2,3)
Problem 1	4	7.9E-4	8.6E-5	6.7E-6	1.8E-3	6.8E-5	3.2E-6	1.6E-7
	8	8.2E-5	6.6E-6	4.5E-7	1.6E-4	6.4E-6	8.8E-7	1.2E-7
	12	2.2E-5	1.5E-6	9.1E-8	3.8E-5	1.5E-6	3.1E-7	6.4E-8
	16	8.6E-6	1.1E-6	1.1E-6	1.6E-5	5.8E-7	2.5E-7	9.3E-8
Problem 2	4	1.6E-4	7.8E-6	1.5E-6	5.5E-5	3.0E-6	1.8E-6*	
	8	9.5E-6	9.3E-8	1.6E-9	1.1E-5	1.7E-8	2.0E-10	3.6E-11
	12	1.8E-6	2.3E-9	9.5E-11	4.4E-6	1.2E-9	4.4E-10	3.6E-11
	16	5.3E-7	1.6E-8	1.7E-8	2.1E-6	1.1E-9	3.5E-10	4.6E-10

CHAPTER 9

APPLICATION OF IDeC TO PARTIAL DIFFERENTIAL EQUATIONS

9.1 Introduction

We now turn our attention to applications of the method of iterated defect correction on the finite difference methods for partial differential equations.

Zadunaisky (1976) applied his technique to partial differential equations for the sole purpose of estimating the global error. In this chapter we attempt to modify his method and to use the resulting algorithm iteratively. Hence the results of this chapter are of an experimental nature.

In the next section we explain the modification of Zadunaisky's principle as applied to parabolic partial differential equations, by introducing certain functions called 'x-splines' and 't-splines'.

Sections 9.3 and 9.4 deal with the application of IDeC respectively to hyperbolic and elliptic partial differential equations. In the case of elliptic equations, we consider IDeC with the aid of piecewise interpolatory polynomials instead of the x-splines and the t-splines used in IDeC for the other two types of partial differential equations.

Computational examples illustrate the performance of the modified IDeC.

9.2 Parabolic Partial Differential Equations

We consider the following initial boundary value problem (IBVP) of parabolic partial differential equations:

$$u_t = f(x,t,u,u_x,u_{xx}), \quad a < x < b, \quad 0 < t \le T$$

$$(9.2.1) \qquad \begin{aligned} u(x,0) &= g_1(x), & a \le x \le b \\ u(a,t) &= g_2(t), & 0 \le t \le T \\ u(b,t) &= g_3(t), & 0 < t \le T. \end{aligned}$$

Let us denote the unique solution of (9.2.1) by $u(x,t)$.

In order to introduce the discretization of (9.2.1) by the method of backward difference (BDM), we choose positive integers N and M, let $h = (b - a)/N$ and $k = T/M$, and define a grid G as

$$(9.2.2) \quad G = \{(x_i, t_j) : x_i = a + ih, \ t_j = jk; \ i = 0(1)N, \\ j = 0(1)M\}.$$

Furthermore, we compute the approximate solution values $u_{i,j+1}$ at the grid-points (x_i, t_{j+1}) by using the finite difference equations:

$$(9.2.3)\quad \frac{u_{i,j+1} - u_{i,j}}{k} - f\Big[x_i, t_{j+1}, u_{i,j+1}, \frac{u_{i+1,j+1} - u_{i-1,j+1}}{2h}, \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{h^2}\Big] = 0, \quad i = 1(1)N-1, \quad j = 0(1)M-1,$$

with the initial and boundary conditions

$$(9.2.4)\quad \begin{aligned} &u_{i,o} = g_1(x_i), \quad i = 0(1)N \\ &u_{o,j} = g_2(t_j) \text{ and } u_{N,j} = g_3(t_j), \quad j = 0(1)M. \end{aligned}$$

Zadunaisky (1976) extimates the resulting global error $u(x_i, t_j) - u_{i,j}$ in the following manner. A <u>bivariate</u> interpolating function $P = P(x,t)$ is found such that $P(x_i, t_j) = u_{i,j}$. Then the NP corresponding to the OP (9.2.1) is taken as

$$(9.2.5)\quad u_t = f(x,t,u,u_x,u_{xx}) + P_t - f(x,t,P,P_x,P_{xx})$$

with the same initial and boundary conditions for the OP. The numerical solution $v_{i,j}$ to the NP (9.2.5) is then found by the same BDM, i.e. using the equations:

$$\begin{aligned} &\frac{u_{i,j+1} - u_{i,j}}{k} \\ &- f[x_i, t_{j+1}, u_{i,j+1}, \frac{u_{i+1,j+1} - u_{i-1,j+1}}{2h}, \\ (9.2.6) \quad &\frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{h^2}] - (P_t)_{i,j+1} \\ &+ f[x_i, t_{j+1}, P_{i,j+1}, (P_x)_{i,j+1}, (P_{xx})_{i,j+1}] = 0, \\ &i = 1(1)N-1, \qquad j = 0(1)M-1. \end{aligned}$$

Since the exact solution to NP (9.2.5) is $P(x,t)$, the global error-$v_{i,j} + P(x_i,t_j)$ in the numerical solution $v_{i,j}$ is known. This known error is taken as the required estimate for the global error.

We observe that in the above method of Zadunaisky an explicit computation of the bivariate interpolating function $P(x,t)$ can be avoided provided that approximate values for the necessary partial derivatives of P are determined only at the points of the grid G. Accordingly we determine univariate interpolating functions which are named as the 'x-splines' and 't-splines'. An x-spline is defined to be a spline function of x which interpolates the data points $(x_i,u_{i,j})$, $i = 0(1)N$ for which j is a fixed integer. Similarly, a t-spline is defined as a spline function of t which interpolates the points $(t_j,u_{i,j})$, $j = 0(1)M$ for which i is a fixed integer. In order to determine approximate values for the derivatives of P appearing in (9.2.6), we merely construct M x-splines and N-1 t-splines, and take the corresponding derivatives of those splines.

We have introduced the above modification in the iterated defect correction method for partial differential equations.

In the case of IBVP (9.2.1) if M is greater than 30, we first improve the solutions in the sub-grid consisting of $m_0 (\leq 30)$ time-steps by an application of IDeC method, and then proceed further in a similar way, making use of the improved solution for the m_0th time step as the starting values for the sub-grid consisting of the next m_0 time-steps. After obtaining improved values in the second sub-grid by another application of IDeC, we proceed to improve the solution values in the next sub-grid till the solution values for Mth time-step are improved.

. Numerical results

In our implementation of IDeC the x-splines and t-splines were taken to be cubic splines (see Tewarson[1980]). We denote by IDeC(i) the IDeC method with 'i' applications of Zadunaisky's principle.

The first five of the following six test problems are from Lindberg (1976), and Problem 6 from Greenspan(1974).

Problems 1 to 3

$$u_t = u_{xx} \qquad 0 < x < \pi, \qquad t > 0$$

$$u(0, t) = 0 = u(\pi, t) \qquad t > 0$$

with the following initial value function $g_1(x)$:

(a) $\qquad g_1(x) = \sin(x) \qquad 0 \leq x \leq \pi$

exact solution: $u(x,t) = e^{-t} \sin x.$

(b) $\qquad g_1(x) = x(\pi - x) \qquad 0 \leq x \leq \pi$

exact solution: $u(x,t) = \frac{8}{\pi} \sum_{n=1}^{\infty} (2n - 1)^{-3} e^{-(2n-1)^2 t} \sin (2n-1)x$

(c) $\qquad g_1(x) = \begin{cases} x & 0 \leq x \leq \pi/2 \\ \pi - x & \pi/2 \leq x \leq \pi \end{cases}$

exact solution: $u(x,t) = \frac{4}{\pi} \sum_{n=1(2)}^{\infty} (-1)^{(n-1)/2} \frac{1}{n^2} e^{-n^2 t} \sin nx$

Problem 4

$$u_t = (2 + \sin u_{xx})u_{xx} + (1 - \sin u)u, \quad 0 < x < \pi/2, \quad t > 0$$

$$u(0,t) = 0, \qquad u(\tfrac{\pi}{2}, t) = e^{-t} \qquad t > 0$$

$$u(x,0) = \sin x \qquad 0 \leq x \leq \pi/2$$

exact solution: $u(x,t) = e^{-t} \sin x.$

Problem 5

Burger's equation

$$u_t = -u\, u_x + u_{xx} \qquad 0 < x < \pi, \qquad t > 0$$

$$u(0,t) = u(\pi,t) = 0 \qquad t > 0$$

$$u(x,0) = \sin x \qquad 0 \le x \le \pi$$

exact solution:

$$u(x,t) = \frac{4\sum_{n=1}^{\infty} e^{-n^2 t}\, I_n(\tfrac{1}{2}) \sin(nx)}{I_0(\tfrac{1}{2}) + 2\sum_{n=1}^{\infty} e^{-n^2 t} I_n(\tfrac{1}{2}) \cos(nx)}$$

where the I_n are the modified Bessel functions.

Problem 6

$$u_t = u_{xx} - x(t-1)e^{-t} \qquad 0 < x < 1, \qquad t > 0$$

$$u(x,0) = 0 \qquad 0 \le x \le 1$$

$$u(0,t) = 0, \qquad u(1,t) = t\, e^{-t} \qquad t > 0$$

exact solution: $u(x,t) = xte^{-t}$.

Table 9.2.1 gives the maximum absolute errors in the numerical solutions obtained when Problems 1 to 3 were solved by the backward difference method (BDM) as well as the IDMC method with $k = .05$ and $h = \pi/10$ and $\pi/20$. The number of points taken for the determination of a t-spline was 20 only.

Table 9.2.1 Maximum Errors by BDM and IDeC with k = .05

		h = $\pi/10$		h = $\pi/20$	
	t	BDM	IDeC(1)	BDM	IDeC(1)
	.05	1.5E-3	3.0E-4	1.2E-3	3.5E-5
	.25	6.3E-3	1.3E-3	5.1E-3	1.6E-4
Problem 1	.45	9.2E-3	1.9E-3	7.5E-3	2.5E-4
	.65	1.1E-2	2.3E-3	9.0E-3	3.2E-4
	.85	1.2E-2	2.5E-3	9.6E-3	3.7E-4
	.05	1.2E-2	1.2E-3	9.9E-3	2.8E-3
	.25	1.5E-2	3.0E-3	1.3E-2	8.6E-4
Problem 2	.45	2.1E-2	4.2E-3	1.7E-2	5.6E-4
	.65	2.7E-2	5.4E-3	2.2E-2	7.1E-4
	.85	3.0E-2	6.3E-3	2.4E-2	8.9E-4
	.05	6.9E-2	2.0E-2	4.1E-2	1.2E-2
	.25	3.0E-2	7.3E-3	1.8E-2	2.3E-3
Problem 3	.45	2.3E-2	4.2E-3	1.4E-2	1.2E-3
	.65	2.1E-2	2.3E-3	1.4E-2	8.4E-4
	.85	2.0E-2	1.1E-3	1.4E-2	5.6E-4

The improvement in the solutions by the IDeC is quite good for each of the three problems solved. It is not worthy that Zadunaisky's principle of error estimation has worked well for Problem 3 for which the initial value function is not smooth whereas the method of Lindberg (1976) failed to estimate the errors reliably. We also observe that the results of IDeC(1) with $h = \pi/10$ are better than those of BDM with $h = \pi/20$. However, IDeC(i) for $i \geq 2$ did not yield better results than IDeC(1).

The backward difference method was employed also to solve Problems 4 and 5. For each time-step, the BDM yielded a system of non-linear equations which was solved by Newton's method. The initial approximation for the method was taken as the solution at the previous time-step, and the iterations were terminated when the maximum change in the values of the solution was less than a tolerance, ATOL.

For each time-step, the system of non-linear equations arising out of the finite difference equations for the neighbouring problem was again solved by Newton's method. The initial guess was taken as the numerical solution of the original problem and iterations were terminated as in the case of the original problem.

Against each of the selected time-steps in Tables 9.2.2 and 9.2.3 are given the maximum error in the solution of Problems 4 and 5 respectively and the error t the central grid-point; also are given the number of iterations required by Newton's method to solve the original problem and the number required when solving a neighbouring problem. The total number m of the time-steps in a sub-grid for an application of IDeC was chosen as 10.

At the mid-points of the time-lines, IDeC(2) produces better results than IDeC(1). However, when accuracy over the entire grid is desired the IDeC(1) method is preferable.

The solution values at points on $x = \frac{1}{2}$ for Problem 6 by the methods BDM and IDeC(3) with $h = .05$ and $k = .1$ and .01 are given in Table 9.2.4.

We notice that the IDeC(3) method with $k = .1$ gives much better results than the BDM with $k = .01$. As for the cost, the number of arithmetic operations required by the IDeC(3) with $k = .1$ was a little less than the number by the BDM with $k = .01$.

We also solved Problem 6 by the BDM and IDeC methods for $h = .05$ and a few values for k. The errors in the computed solutions were measured by the Euclidean norm in the grid comprising the points (x,t) with $x = 0(.05)1$ and $t = 0(.1)1$. The results in Table 9.2.5 indicate that in this case our implementation of the IDeC method is quite efficient.

Table 9.2.2 Results for Problem 4 with $h = \pi/20$, $k = .1$ and XTOL = .0001

	errors over time-line				errors at centre	
time-step	BDM	IDeC(1)	No. of iterations OP	 NP	BDM	IDeC(1)
1	3.2E-3	9.0E-4	7	4	2.9E-3	2.9E-4
5	6.5E-3	2.3E-3	8	6	6.3E-3	4.9E-4
10	4.2E-3	5.8E-4	6	4	4.2E-3	5.5E-4
15	2.3E-3	6.5E-4	8	5	2.3E-3	3.6E-4
20	1.3E-3	1.9E-4	8	4	1.3E-3	9.5E-5
25	6.6E-4	2.1E-4	7	4	6.6E-4	1.2E-4
30	4.4E-4	9.0E-5	6	3	4.4E-4	8.6E-6
35	2.8E-4	9.6E-5	6	3	2.8E-4	7.7E-5

Table 9.2.3 Results for Problem 5 with h = $\pi/10$, k = .1 and XTOL = .0001

	errors over time-line				errors at centre	
time-step	BDM	IDeC(1)	No. of iterations		BDM	IDeC(1)
			OP	NP		
1	1.1E-2	3.1E-3	3	2	4.9E-3	6.5E-4
5	2.0E-2	3.9E-3	3	2	1.9E-2	3.3E-5
10	2.2E-2	2.1E-3	2	2	2.2E-2	6.4E-4
15	2.0E-2	1.4E-3	2	2	2.0E-2	7.9E-4
20	1.6E-2	9.2E-4	2	2	1.6E-2	7.8E-4
25	1.2E-2	6.8E-4	2	2	1.2E-2	6.4E-4
30	8.9E-3	4.9E-4	2	2	8.9E-3	4.8E-4
40	4.5E-3	2.5E-4	2	2	4.5E-3	2.5E-4
55	1.4E-3	8.1E-5	2	1	1.4E-3	8.1E-5

Table 9.2.4 Problem 6. Solution values when $x = \frac{1}{2}$.

t	BDM k = .1	BDM k = .01	IDeC(3) k = .1	IDeC(3) k = .01	exact solution
.2	.07808528	.08143904	.08178577	.08187299	.08187307
.4	.13029526	.13367796	.13405136	.13406401	.13406401
.6	.16166281	.16435400	.16464544	.16464349	.16464349
.8	.17754286	.17952423	.17973062	.17973159	.17973159
1.0	.18239586	.18379544	.18391273	.18393972	.18393972

Table 9.2.5 Errors obtained by the BDM and IDeC methods with h=.05

k	BDM	IDeC(1)	IDeC(2)	IDeC(3)
0.1000	0.6196	0.1648	0.0867	0.0678
0.0500	0.2911	0.0380	0.0073	0.0047
0.0100	0.1993	0.0202	0.0045	0.0024
0.0025	0.1712	0.0191	0.0033	0.0031
0.0010	0.0631	NC	NC	NC

9.3 Hyperbolic Partial Differential Equations

First, we consider the initial boundary value problem

$$u_t = c\ u_x \qquad c > 0, \qquad 0 \leq x \leq 1, \qquad 0 \leq t \leq T$$

$$(9.3.1) \qquad u(x,0) = g_1(x) \qquad 0 \leq x \leq 1$$

$$u(1,t) = g_2(x) \qquad 0 \leq t \leq T.$$

In the notation of Section 9.2, the 'leap-frog' method for (9.3.1) is given by the equations

$$\frac{u_{i,j+1} - u_{i,j-1}}{2k} - c\,\frac{u_{i+1,j} - u_{i-1,j}}{2h} = 0$$

subject to the conditions

$$u_{i,o} = g_1(x_i), \qquad i = 0(1)N$$

$$u_{N,j} = g_2(t_j), \qquad j = 0(1)M.$$

We consider the following computational example of a non-linear problem.

Problem 1

$$u_t = -\frac{\partial}{\partial x}\left(\tfrac{1}{2}u^2\right) \qquad 0 < x < 1, \qquad t > 0$$

$$u(x,o) = 1 - x \qquad 0 < x < 1$$

$$u(1,t) = 0 \qquad t > 0$$

exact solution: $u(x,t) = (1 - x)/(1 + t)$.

We solved Problem 1 by the leap-frog method

IDeC method with h = .05 and k = .015. The boundary

on $x = 0$ were taken as $u(0,t) = 1/(1 + t)$, and the

values were calculated from the Lax-Wendroff scheme.

$$u_{i,1} = u_{i,o} + \tfrac{1}{2}(u_{i+1,o} - u_{i-1,o})k/h$$

$$+ \tfrac{1}{2}(u_{i+1,o} - 2u_{i,o} + u_{i-1,o})k^2/h^2.$$

The maximum absolute errors for the tenth ti

by the leap-frog, IDeC(1) and IDeC(2) methods were

9.1E-6 and 4.7E-6 respectively.

Next, we consider the IBVP:

$$
\begin{aligned}
& u_{xx} = u_{tt} && 0 < x < a, \quad t > 0 \\
& u(x,0) = f_1(x) && 0 \le x \le a \\
(9.3.2) \qquad & u_t(x,0) = f_2(x) && 0 < x < a \\
& u(0,t) = g_1(t) && t \ge 0 \\
& u(a,t) = g_2(t) && t \ge 0.
\end{aligned}
$$

With the notation of Section 9.2, the numeric

solution $u_{i,j}$ to the above IBVP may be found by th

three-level implicit' scheme:

$$\frac{1}{h^2}[\lambda\delta_i^2 u_{i,j+1} + (1 - 2\lambda)\delta_i^2 u_{i,j} + \lambda\delta_i^2 u_{i,j-1}]$$

$$= \frac{1}{k^2}\delta_j^2 u_{i,j},$$

where λ is a 'relaxation' factor and $\delta_i^2 u_{i,j}$ is the operator

$$\delta_i^2 u_{i,j} = u_{i+1,j} - 2u_{i,j} + u_{i-1,j}.$$

For our choice of $\lambda = \frac{1}{4}$, the above implicit scheme becomes stable for all choices of h and k and requires the solution of a tridiagonal system of equations, for each time-step. (For details, see Ames[1977] and Greenspan[1974]).

Problems 2 to 4 were solved by the implicit scheme mentioned above with the necessary modifications, as in pp 174-175 of Greenspan' (1974), for Problem 4.

Problem 2

$$u_{xx} = u_{tt}$$

$$u(x,0) = 2x + e^x, \qquad u_t(x,0) = -e^x$$

$$u(0,t) = e^{-t}, \qquad u(1,t) = 2 + e^{1-t}$$

exact solution: $u(x,t) = 2x + e^{x-t}$.

Problem 3

$$u_{xx} = u_{tt}$$

$$u(x,0) = \sin \pi x \ , \qquad u_t(x,0) = 0$$

$$u(0,t) = 0 = u(1,t)$$

exact solution: $u(x,t) = \sin \pi x \cos \pi t$.

Problem 4

$$u_{xx} = u_{tt} + 2(t-x)(t+x+2)u^3$$

$$u(x,0) = \frac{1}{1+x}, \qquad u_t(x,0) = -\frac{1}{1+x}$$

$$u(0,t) = \frac{1}{1+t}, \qquad u(1,t) = \frac{1}{2(1+t)}$$

Exact solution: $u(x,t) = \frac{1}{(1+x)(1+t)}$

In Table 9.3.1 we give the errors, by the Euclidean norm, in the computed solutions obtained by the implicit and IDeC methods. We took h = .025 and k = .01 and/or .1, and the grid was bounded by t = 1.

Table 9.3.1 Errors by the IDeC methods with h = .025

Problem	k	implicit	IDeC(1)	IDeC(2)
2	.01	7.1E-2	5.7E-2	5.4E-2
3	.1	1.2E-0	1.1E-0	9.1E-1
	.01	2.9E-1	1.6E-1	NC
4	.01	1.8E-0	1.4E-0	1.0E-0

9.4 Elliptic Partial Differential Equations

We now consider the Poisson's equation

$$u_{xx} + u_{yy} = g(x,y) \qquad 0 \leq x,y \leq 1 \tag{9.4.1}$$

with $u(x,y) = f(x,y)$ on the boundary of the unit square.

In the notation of Section 9.2 with $t = y$, $T = 1$ and $k = h$, the five-point finite difference scheme for (9.4.1) is given by

$$(9.4.2) \qquad u_{i-1,j} + u_{i+1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}$$

$$= h^2 g(x_i, y_j), \qquad i,j = 1(1)N-1.$$

The above system of linear equations together with the discrete boundary conditions is solved iteratively by the SOR method. The iterations were terminated when the difference in two successive iterates is less than TOL, measured in the one-norm

The NP for (9.4.1) is

$$u_{xx} + u_{yy} = P_{xx} + P_{yy} \qquad 0 \leq x,y \leq 1$$

with the meaning of P as given in Section 9.2. The finite difference equations for this NP are the same as (9.4.2) but with the righthand side $h^2[(P_{xx})_{i,j} + (P_{yy})_{i,j}]$. The approximate values for the derivatives P_{xx} and P_{yy} at (x_i, y_j) are now found using piecewise interpolatory polynomials of degree m which are defined analogous to the x-splines and the t-splines of Section 9.2.

We took three problems of the form (9.4.1) with $f = u$ and g as given below. The maximum absolute errors in their solutions obtained by the five-point scheme and the IDeC method using univariate polynomials of degree 8 are tabulated in Table 9.4.1.

Problem 1

$$g(x,y) = -2[x(1-x) + y(1-y)]$$

exact solution: $u(x,y) = x(1-x)y(1-y)$.

Problem 2

$$g(x,y) = 4/(x^2+y^2+1)^2$$

exact solution: $u(x,y) = \ln(x^2+y^2+1)$.

Problem 3

$$g(x,y) = 0$$

exact solution: $u(x,y) = e^{\pi x} \cos \pi y$.

Table 9.4.1 Errors by the 5-point and IDeC methods

		h = 1/16			h = 1/16		
Problem	TOL	Five-point	IDeC(1)	IDeC(2)	Five-point	IDeC(1)	IDeC(2)
1	.01	5E-4	7E-5	5E-4	3E-4	4E-5	5E-4
2	.001	4E-4	1E-5	4E-6	8E-5	1E-6	4E-5
3	.01	4E-2	1E-3	4E-2	1E-2	8E-4	2E-2

The results show that there is a slight improvement in the accuracy of the solutions only for a single application of defect correction.

9.5 Conclusion

In this chapter we have introduced a slight modification in the construction of neighbouring problems used in the defect correction procedure, and applied the modified IDeC to some classes of problems in partial differential equations.

In the modified IDeC applicable to partial differential equations we need to construct global approximations as functions of a single variable in contrast to functions involving more than one variable, used by Zadunaisky (1976).

Computational results indicate that the improvement brought about by the modified IDeC is sometimes substantial, and that the first iterate of this IDeC is usually more accurate than the other iterates.

CHAPTER 10

CONCLUSIONS

The primary objective of the work presented in this thesis has been to apply the method of IDeC in new ways and to new types of operator equations. Consequently, we have applied IDeC to integral equations; presented a variant of IDeC called Successive Updated IDeC; and tested the performance of a modified version of IDeC applicable to partial differential equations (PDEs). Moreover, we have applied IDeC on extrapolation (IDeCE), extrapolation on IDeC (SEIDeC), IDeC on the deferred correction process of Fox (IDDeC), and extrapolation on IDDeC (SEIDDeC). We have also compared our new methods of iterative improvement with other methods of iterative improvement.

10.1 Discussion of Numerical Results

10.1.1 Applicability of IDeC

Fredholm Integral Equations

We have applied IDeC on quadrature methods for second kind FIEs. We have considered $IDeC_T$ and $IDeC_S$ methods based respectively on trapezoid and Simpson's rule. The following statements have been found to be true on the basis of our theoretical and numerical results (see

Figure 3.2.1 and tables in Sections 2.2.2 and 3.2).

(1) The higher the order n of the basic quadrature method, the more rapid the convergence of the corresponding ILeC.

(2) More specifically, the increase in order of convergence of ILeC per iteration is equal to n. However, the maximum attainable order of ILeC is independent of n but depends on the degree m of polynomials used.

(3) For a given m, the iterates of ILeC_T and ILeC_S converge to the same solution but the latter requires fewer iterations than the former.

(4) The greater the degree m ($4 \leq m \leq 12$), the higher the accuracy in the solution produced by ILeC.

ILeC does not achieve an order greater than the maximum attainable stated by the theory, except for a single case (Problem 3 of Section 2.2.2). This is due to the fact that the polynomials which extend the discrete approximate solution do not introduce an error due to interpolation (since the exact solution of the problem is a polynomial of degree one).

Nonlinear Volterra Integral Equations

Our computational results for IDeC techniques IDeC_M, IDeC_T and IDeC_N based on the midpoint rule, the trapezoid rule, and Noble's method respectively, show the validity of the statements (1) and (4) mentioned above for FIEs. Thus IDeC$_N$ yields more accurate solution than IDeC$_T$, and IDeC$_T$ more accurate solution than IDeC$_M$. Our IDeC$_T$ method turns out to be better than Garey's modified increment methods and Netravali's spline approximation method for nonlinear VIEs.

We have also found that for a given stepsize, the IDeC techniques generally produce more accurate solution than any other discretization method.

Linear Volterra Integral Equations

The IDeC method based on quadrature method that uses the trapezoid rule for the solution of linear VIEs produces as good results as the IDeC$_T$ method for nonlinear VIEs (Tables 2.4.1 and 8.2).

10.1.2 Applicability of SEIDeC

In the case of FIEs, the numerically estimated order of the SEIDeC method has been found to agree with the theoretically predicted order (see equation (3.3.3) and Tables 3.4.1 and 3.4.2).

10.1.3 Applicability of IDDeC

Fredholm Integral Equations

The computed orders of convergence for the iterations of the IDDeC method based on Gregory's formula for FIEs agree with the theoretical orders (see equation (4.3.19a) and Tables 4.4.4 to 4.4.7).

We have also pointed out that IDLeC based on trapezoid rule with correction is easier to implement than IDIeC based on Simpson's rule with correction.

Volterra Integral Equations

The local approach of ILC has been found to be superior to the global approach, and IDDeC based on the local ILC better than the one based on the global ILC. However in the case of nonlinear VIEs, our results show that the IDLeC based on the global IDC yields results almost as accurate as the IDDeC based on the local ILC, even though the iterates of the global IDC fail to converge (see Table 4.4.14b).

Remarks

In the case of VIEs, whether linear or nonlinear, numerical results obtained by II eC and IDI eC methods do not indicate any asymptotic behaviour concerning the order of convergence of the methods.

A comparative study of our methods with certain other methods of iterative improvement has been made in Chapter 8.

For linear VIE (2.4.1), we also implemented the ILeC which makes use of the following idea of error estimation suggested by Frank (see Section 6, Frank[1976]): discretize the equation

$$y(x) - \int_0^x k(x,t)y(t)dt = 2f(x) - P_m(x) + \int_0^x k(x,t)P_m(t)dt$$

and take the results directly as better approximation to $y(x_1)$. In our experiments, this modification produced the same results as the IDeC based on Zadunaisky's idea.

10.2 Conclusions

Our experiments show that for the construction of perturbations or defects in IDeC, the use of piecewise polynomial interpolation is far more rewarding than that of cubic spline interpolation.

For solving second kind FIEs, each of the methods IDeC, SEIDeC, IDDeC and SEIDDeC is very powerful and effective in practice. However, the method IDDeC/SEIDDeC based on the trapezoid rule or the method SEIDeC based on Simpson's rule is to be chosen for use in general purpose routines.

In the case of general purpose routines for solving VIEs, the IDDeC method based on the local approach of IDC which uses Gregory formula is to be preferred.

When moderately accurate solutions are desired at low costs, our newly introduced methods come in handy.

In the case of integral equations, standard software is available only for second kind linear FIEs with smooth kernels and Green's function kernels (Delves, et al[1981]).

But our work indicates that the methods such as IDeC, SEILeC, IDDeC and SEIDDeC will be of use in the development of standard general purpose routines for the solution of integral and for other operator equations.

Among the methods of iterative improvement, techniques based on IDeC are expected to be prominent in the present time of emphasis on mathematical software for large computers.

10.3 Future Work

Finally our work on methods of iterative improvement points the way for future research in numerous new areas. Some of these are the following.

(a) Implementation of ILeC to nonlinear FIEs and integro-differential equations.

(b) Applicability of SEILeC to nonlinear FIEs and ordinary differential equations.

(c) Implementation of SEIDeC on simple numerical methods for integro-differential equations, with the use of the results of extrapolation methods (e.g. Chang[1982]).

(d) Application of IDLeC and SEIDDeC to ordinary and partial differential equations.

(e) Implementation of IDDeC based on Pereyra's version of deferred correction.

(f) Application of SUILeC with the use of interpolating polynomials of varying degrees to initial value problems of ordinary differential equations.

During the final stages of our work, we came across a new type of iterative improvement technique of Böhmer[1981] called by him the Discrete Newton Method (DNM). This method is obtained as a combination of a Newton and a discretization Method. It produces approximations of orders 2n, 3n, ..., where n is the order of accuracy of the initial numerical solution. Böhmer also presents a general theory for DNMs, IDeCs and IDCs applicable to any discretization method with an asymptotic expansion. DNM is superior to and cheaper than IDeC and IDC (Pereyra's version) for problems requiring solutions of nonlinear systems of equations.

In future we would like to compare the efficiency of our methods with DNMs taking into consideration parameters such as CPU time, storage and number of function evaluations.

LIST OF REFERENCES

Alfeld, P.[1975] 'A survey of Zadunaisky's device applied to ordinary Differential Equations', M.Sc. Dissertation, Univ. of Dundee, Scotland.

Ames, W.F.[1977] 'Numerical Methods for Partial Differential Equations', 2nd Ed., Academic Press, New York.

Baker, C.T.H.[1977] 'The Numerical Treatment of Integral Equations', Clarendon Press, Oxford.

Böhmer, K.[1981] 'Discrete Newton methods and iterated defect corrections', Numer. Math. 37, 167-192.

Chang, S.H.[1982] 'On certain extrapolation methods for the numerical solution of integro-differential equations', Math. Comp. 39, 165-171.

Daniel, J.W. and Martin, A.J.[1977] 'Numerov's method with deferred corrections for two-point boundary value problems', SIAM J. Numer. Anal. 14, 1033-1050.

Davis, H.T.[1962] 'Introduction to Nonlinear Differential and Integral equations', Dover, New York.

Delves, L.M. and Walsh, J.[1974] 'Numerical Solution of Integral Equations', Clarendon Press, Oxford.

Delves, L.M., Abd-Elal, L.F., and Hendry, J.A.[1981] 'A set of modules for the solution of integral equations', Comp. J. 24, 184-190.

Fox, L.[1947] 'Some improvements in the use of relaxation methods for the solution of ordinary and partial differential equations', Proc. Roy. Soc. A190, 31-59.

Fox, L.[1949] 'The solution by relaxation methods of ordinary differential equations', Proc. Camb. Phil. Soc. 45, 50-68.

Fox, L.[1950] 'The numerical solution of elliptic differential when the boundary conditions involve a derivative', Phil. Trans. A242, 345-378.

Fox, L.[1957] 'The Numerical Solution of Two-Point Boundary Value Problems in Ordinary Differential Equations', Oxford University Press.

Fox, L.[1962] 'Numerical Solution of Ordinary and Partial Differential Equations', Pergamon Press, Oxford.

Fox, L.[1977] 'Finite-difference methods for elliptic boundary value problems' in D. Jacobs[1977], 799-881.

Frank, R.[1975] 'Schalzungen des globalen Diskretisierungsfehlers bei Runge-Kutta-Methoden', ISNM 27, S. 45-70, Birkhauser, Basel.

Frank, R. and Ueberhuber, C.W.[1975] 'Iterated defect correction for Runge-Kutta methods', Report No. 14/75, Inst. f. Num. Math., Technical Univ. of Vienna.

Frank, R.[1976, 1977] 'The method of iterated defect-correction and its application to two-point boundary value problems', Part I : Numer. Math. 25, 409-419
Part II: Numer. Math. 27, 407-420.

Frank, R. and Ueberhuber, C.W.[1977] 'Iterated defect correction for the efficient solution of stiff systems of ordinary differential equations', BIT 17, 146-159.

Frank, R., Hertling, J., and Ueberhuber, C.W.[1976] 'Iterated defect correction based on estimates of the local discretization error', Report No. 18/76, Inst. f. Num. Math., Technical Univ. of Vienna.

Frank, R., Hertling, J., and Ueberhuber, C.W.[1977] 'An extension of the applicability of iterated deferred corrections', Math. Comp. 31, 907-915.

Garey, L.[1975] 'Solving nonlinear Second Kind Volterra equations by modified increment methods', SIAM J. Numer. Anal. 12, 501-508.

Gragg, W.B.[1965] 'On extrapolation algorithms for ordinary initial value problems', SIAM J. Numer. Anal. 2, 384-403.

Grant, J.A.[1969] 'Derivation of correction terms for general quadrature formula', Proc. Cambridge Philos. Soc. 66, 571-589.

Greenspan, D.[1974] 'Discrete Numerical Methods in Physics and Engineering', Academic Press, New York.

Hock, W.[1981] 'An extrapolation method with stepsize control for nonlinear Volterra integral equations', Numer. Math. 38, 155-178.

Jacobs, D.[1977](ed)'The State of the Art in Numerical Analysis', Academic Press, New York.

Kershaw, D.[1974] 'Volterra equations of the second kind', in Delves and Walsh[1974], 140-161.

Lambert, J.D.[1977] 'The initial value problem for ordinary differential equations', in D. Jacobs[1977], 451-500.

Lentini, M. and Pereyra, V.[1975] 'Boundary Value problem solvers for first order systems based on deferred corrections', in A.K. Aziz (Ed.) 'Numerical Solution of Boundary Value problems for Ordinary Differential Equations', Academic Press, New York.

Lindberg, B.[1976] 'Error estimation and iterative improvement for the numerical solution of operator equations', UIUCDCS-R-76-820, Dept. of Comp. Science, Univ. of Illinois at Urbana-Champaign.

Mayers, D.F.[1974] 'Quadrature methods for Fredholm equations of the second kind', in L.M. Delves and J. Walsh[1974], 64-79.

Netravali, A.N.[1973] 'Spline approximation to the solution of the Volterra integral equation of the second kind', Math. Comp. 27, 99-105.

Netravali, A.N. and de Figueiredo, R.J.P.[1974] 'Spline approximation to the solution of the linear Fredholm equation of the second kind', SIAM J. Numer. Anal. 11, 538-549.

Noble, B.[1969] 'Instability when solving Volterra integral equations ofthe second kind by multistep methods', Lecture Notes in Mathematics 109, Springer-Verlag, Berlin, 23-29.

Noble, B. [1977] 'the numerical solution of integral equations' in Jacobs [1977], 915-966

Pereyra, V.[1966] 'On improving an approximate solution of a functional equation by deferred corrections', Numer. Math. 8, 376-391.

Pereyra, V.[1967a] 'Iterated deferred corrections for nonlinear operator equations', Num. Math. 10, 316-323.

Pereyra, V.[1967b] 'Accelerating the convergence of discretization algorithms', SIAM J. Numer. Anal. 4, 508-533.

Pereyra, V.[1968] 'Iterated deferred corrections for nonlinear boundary value problems', Num. Math. 11, 111-125.

Pereyra, V.[1970] 'Highly accurate numerical solution of casilinear elliptic boundary value problems in n dimensions', Math. Comp. 24, 771-783.

Pereyra, V.[1973] 'High order finite difference solution of differential equations', Report STAN-CS-73-348, Comp. Science Dept., Stanford University.

Richardson, L.F.[1910] 'The approximate arithmetical solution by finite differences of physical problems involving differential equations', Phil. Trans. A210, 307-357.

Richardson, L.F.[1927] 'The deferred approach to the limit. I: Single lattice', Philos. Trans. Roy. Soc. London, Ser. A226, 299-349.

Sathiaraj, D.D. and Sankar, R.[1982] 'The method of successive updated iterated defect correction and its application to second kind Volterra integral equations', BIT 22, 390-394.

Sathiaraj, D.D. and Sankar, R.[1983] 'The method of successive extrapolated iterated defect correction and its application to second kind Fredholm's integral equations', BIT (in press).

Skeel, R.D.[1980] 'A theorectical framework for proving accuracy results for deferred corrections' UIUCDCS-F-80-892, Dept. of Comp. Science, Univ. of Illionois at Urbana-Champaign.

Stetter, H.J.[1965] 'Asymptotic expansions for the error of discretization algorithms for nonlinear functional equations', Numer. Math. 7, 18-31.

Stetter, H.J.[1974] 'Economical global error estimation', in R.A. Willoughby[1974], 245-258.

Stetter, H.J.[1978] 'The defect correction principle and discretization methods', Numer. Math. 29, 425-443.

Stetter, H.J.[1979] 'Global error estimation in Adams-PC Codes', ACM Trans. Math. Software 5, 415-430.

Tewarson, R.P.[1980] 'On the use of splines for the numerical solution of nonlinear two-point boundary value problems', BIT 20, 223-232.

Walsh, J.[1974] 'Quadrature and Fourier Series', in L.M. Delves and J. Walsh[1974], 12-22.

Willoughby, T.A.[1974] (Ed.) 'Stiff Differential Systems', Plenum Press, New York and London.

Zadunaisky, P.E.[1966a] 'A method for the estimation of errors propagated in the numerical solution of a system of ordinary differential equations', in Proc. Intern. Astron. Union, Symposium 25, Thessaloniki, 1964, Academic Press, New York.

Zadunaisky, P.E.[1966b] 'The motion of Halley's comet during the return of 1910', Astr. J. 71, 20-27.

Zadunaisky, P.E.[1970] 'On the accuracy in the numerical computation of orbits', in 'Periodic Orbits, Stability and Resonances' Symposium at the Univ. of Sao Paulo, Brazil 1969, D. Reidel Pub. Co., Dordrecht.

Zadunaisky, P.E.[1972] 'On the determination of non-gravitational forces acting on comets', in Proc. Intern. Astron. Union, Symposium No. 45, Leningrad, USSR, 1970, D. Reidel Pub. Co,., Dordrecht.

Zadunaisky, P.E.[1976] 'On the estimation of errors propagated in the numerical integration of ordinary differential equations', Numer. Math. 27, 21-40.

Zadunaisky, P.E. and Lafferriere, G.[1980] 'On an iterative improvement of the approximate solution of some ordinary differential equations', Comp. & Maths with Appls. 6, 147-154.